The MCM at 21: COMAP's Mathematical Contest in Modeling

published by

The Consortium for Mathematics
and Its Applications, Inc.
Suite 210
57 Bedford St.
Lexington, MA 02173

edited by

Paul J. Campbell
Campus Box 194
Beloit College
700 College St.
Beloit, WI 53511–5595
campbell@beloit.edu

ISBN 1–933223–04–9
Typeset and printed in the U.S.A., using LaTeX and Apple Macintosh computers.

Table of Contents

The MCM Comes of Age

Paul J. Campbell
Dept. of Mathematics and Computer Science
Beloit College
Beloit, WI 53511
campbell@beloit.edu

This special volume commemorates the second ten years of the Mathematical Contest in Modeling (MCM). With the 2005 contest, the MCM turned 21.

The idea for the MCM came from Ben Fusaro in late 1983, who envisioned an "applied" analogue of the famous annual William Lowell Putnam Contest in mathematical problem-solving. Sol Garfunkel, Executive Director of COMAP, realized the potential for the idea and secured grant funding.

Since the first contest, participation has grown steadily, with the 2005 MCM involving 828 teams of undergraduates, from more than 300 institutions in 11 countries.

Moreover, the MCM inspired expansion in 1999 to the Interdisciplinary Contest in Modeling (ICM), which in 2005 involved an additional 164 teams from 4 countries. In a few more years, we expect to celebrate the ICM's tenth anniversary with a volume like this one and its predecessor *UMAP Models: Tools for Teaching 1994*, which celebrated the tenth anniversary of the MCM. That volume is available on COMAP's special Modeling Resource CD-ROM (http://www.comap.com/product/?idx=613).

Short articles in this volume give the current rules of the MCM, describe the process of judging the solution papers, and offer assorted short tips from advisors of teams. We also reprint here from the earlier volume Ben Fusaro's recounting of the history, background, and emergence of the MCM.

This volume contains all of the 20 problems set in the second ten years of the MCM. For each year, one Outstanding paper is included, together with abbreviated accompanying commentaries.

We also include a digest of the current contest rules and an account of how it is judged, by contest director Frank Giordano.

Coaches of multiple Outstanding teams offer tips on how to help teams succeed. Anne Dougherty emphasizes the importance of student ability, training, and team preparation. James Morrow notes the importance of publicity of successes in attracting students to the contest and to major in mathematics and highlights the excitement of the participants. Garrett Mitchener offers the unique perspective of someone who was a three-time Outstanding winner as a student and now coach of an Outstanding team in 2005; his theme is to "go with what you know" in terms of the mathematics, the computing support, and the writing. Edward Allen and colleagues comment on how to make the contest fun and a celebration.

Finally, we collect together advice from the judges over the years, who have a lot to say about what to put in—and what to leave out of—a contest paper.

A few Outstanding papers have used highly sophisticated mathematics, such as neural nets or genetic algorithms. Often, however, simple mathematics may suffice to realize great insight. Above all, as Prof. Fusaro remarks in his account of background and history, the value of the MCM lies not in the competition but in the educational experience of students.

We hope that you enjoy this volume. If your institution has not yet participated in the MCM, you may obtain information and register for the next MCM or ICM contest (both in February) by going to `http://www.comap.com/undergraduate/contests/` .

About the Editor

Paul Campbell graduated summa cum laude from the University of Dayton and received an M.S. in algebra and a Ph.D. in mathematical logic from Cornell University. He has been at Beloit College since 1977. He is Reviews Editor for *Mathematics Magazine* and has been editor of *The UMAP Journal* since 1984, a responsibility that he enjoys immensely and helps him be even more of a "generalist." He is co-author of COMAP's applications-oriented collegiate mathematics text *For All Practical Purposes* (7th ed., W.H. Freeman, 2006), already used by more than half a million students.

Outstanding MCM Teams 1985–1994

	1985–1994	**1995–2004**
Beloit College, WI	1991, 1994	
Bethel College, MN		2001, 2002
California Institute of Technology, CA	1989	2003
California Polytechnic State Univ., CA	1989, 1990	2000
Calvin College, MI	1985, 1987	1997
Colorado School of Mines, CO	1986	
Cornell University, NY	1993	
Donghua University, Shanghai, China		2003
Drake University, IA	1988, 1989	
Duke University, NC		1998, 1999, 2000, 2001, 2002
East China University of Science and Technology, Shanghai		1997
Eastern Oregon University, OR		1998
Fudan University, China		1996
Georgetown University, DC	1986	
Gettysburg College, PA		1996
Grinnell College, IA	1986	1994
Harvard University, MA	1988	1997, 2003, 2004
Harvey Mudd College, CA	1985, 1986 (2), 1989	1995 (2), 1998 (2), 1999 (3), 2001, 2002, 2003, 2004
Hiram College, OH	1991	
Humboldt State University, CA	1990	
Iowa State University, IA		1995
Lawrence Technological University		2001
Lewis & Clark College, OR		2000
Macalester College, MN		1995, 1997, 1998
Maggie Walker Governor's School, VA		2000, 2001, 2002
Merton College, Oxford University, U.K.		2004
Moorhead State University, MN	1987	
Mount St. Mary's College, MD	1985, 1991	
National University of Defence Tech., Changsha, China		2000
Nazareth College, NY	1993	
New Mexico State University, NM	1985	
North Carolina School of Science and Mathematics, NC	1988, 1989, 1992, 1994	1995, 1999, 2002
Ohio State University, OH	1989	
Oklahoma State University, OK	1992	
Pacific Lutheran University, WA	1999	

Peking University, Beijing, China		2003
Pomona College, CA	1986, 1992	1996, 1997
Rensselaer Polytechnic Institute, NY	1987	
Ripon College, WI	1991	
Rose-Hulman Institute of Tech., IN	1990	1997, 1999
St. Bonaventure University, NY		1996
Southeast University, Nanjing, China		2003
Southeast Missouri State Univ., MO		1995
Southern Methodist University, TX	1985	
Southern Oregon State University, OR	1990	
Stellenbosch University, South Africa	2001	
Stetson University, FL		1998
Tsinghua University, Beijing, China		1998
U.S. Air Force Academy, CO	1990	
U.S. Military Academy, NY	1988, 1993	2000, 2001, 2002
University College Cork, Cork, Ireland		2001, 2004
University of Alaska Fairbanks, AK	1990, 1991, 1993	1995, 1997, 1999
University of Calgary, Alberta, Canada	1994	
University of California—Berkeley, CA	1986, 1993	1999
University of Colorado—Boulder, CO	1992	2000, 2002, 2003, 2004 (2)
University of Colorado—Denver, CO	1987	
University of Dayton, OH	1989	
University of North Carolina, NC	1994	1996
University of Puget Sound		1999
University of Science and Technology of China, Hefei		1996
University of Toronto, Ontario, Canada	1986, 1988, 1994	1997
University of Washington, WA		2002, 2003 (2), 2004
University of Western Ontario, Ontario, Canada	1991	
Wake Forest University, NC		1996, 2000, 2001, 2002 (2)
Washington University, MO	1985, 1986, 1989, 1992	1996, 1997, 2000
Worcester Polytechnic Institute, MA		1996
Youngstown State University, OH		2003
Zhejiang University, Hangzhou, China		2003

Background and History of the MCM

Bernard (Ben) A. Fusaro
Dept. of Mathematics
Florida State University
Tallahassee, MD 21801
fusaro@math.fsu.edu

Genesis

The concept of a national applied mathematics contest for undergraduates occurred to me in October 1983. The idea surfaced because of difficulties that we were having at Salisbury State University (now Salisbury University) with getting our students to prepare for the Putnam Mathematical Competition.

Salisbury has a high percentage of first-generation college students, and they tend to view facing such a formidable exam as an ordeal. The practice of the Putnam of reporting a large proportion of low numerical scores adds to the chilling effect. Finally, the small amount of applied content of Putnam problems did not generate much enthusiasm in practical-minded students.

There was much more to my notion of an applied mathematics contest than just offering different questions that would merit higher scores. For a dozen years, I had chafed at the overemphasis in established mathematics of the pure, formalistic approach, almost devoid of content. On many campuses, there was scarcely any appreciable applied or constructive mathematics presence.

In my mind, (classical) applied mathematics, computational mathematics, and statistics are as important a part of contemporary mathematical activities and curricula as pure mathematics. The model that I had in mind represents each of these four as a vertex of a tetrahedron. The edges, faces, and interior represent activities such as applied linear algebra, numerical analysis, or operations research. The Putnam deals with a small neighborhood of the pure mathematics vertex at the lofty apex of the tetrahedron. It would be difficult to tell from Putnam questions that the computer even existed.

These thoughts merged and then popped up in verbal form as "Applied Putnam." My on-campus colleagues liked the idea, but it seemed prudent to check with some off-campus mathematicians who had long involvements in applied mathematics. Calls to M.S. Klamkin (University of Alberta), H.O. Pollak (Bell Labs), and E.Y. Rodin (Washington University) elicited favorable responses and encouragement to proceed. I then called A.P. Hillman, who has had many years of experience with the Putnam. He urged me to start with a small pilot project and warned that I might be underestimating the difficulty of starting a national contest. (He was right.)

Being Chair of the Education Committee of the Society for Industrial and Applied Mathematics (SIAM) gave me a natural forum for this project. I sent an outline of a proposal for a pilot contest to the committee in November 1983. The gist of the proposal is illustrated in **Table 1**.

Table 1.
The original proposal for an "Applied Putnam."

	Pure Putnam	"Applied Putnam"
Time of contest	December	March
Sessions	Two (three hours each)	Two (three hours each)
Number of problems	12	2
Type of problems	Structural, pure	Contextual, applied
Format	Individual students No calculator or computer aids	Teams of three students Microcomputers allowed

The committee liked the proposal but had strong reservations about the time allotted per problem. The feeling was that an applied mathematics problem could not be done in half a day; estimates ran from a day to a week. One experienced SIAM officer said that a realistic problem would need a whole semester! These observations, coupled with my own fairly unshakable view that a contest for undergraduates should not occupy more than a weekend, doomed one of my favorite schemes, that teams should be required to do one continuous problem and one discrete problem.

Although the committee looked on the idea with favor, SIAM's leadership felt that the committee already had enough projects and that it should continue to concentrate on the K–12 level. However, so many people had judged the idea to be good and workable that I decided to seek another forum.

Funding

Warren Page, then editor of the *College Mathematics Journal*, gave an invited lecture to the Maryland–DC–Virginia Section of the Mathematical Association of America in November of 1983. His lively presentation included many applied examples, so I approached him after his talk. Page listened as attentively as he could, while being badgered by another mathematician who kept trying to tie Page's talk to the Vietnam War. Page's initial reaction was that the concept was interesting but not feasible.

About three weeks later, Page called me at home to say that he had given this concept of an applied contest quite a bit of thought; it was a valuable idea and it ought to be done. Moreover, he had broached the subject to Sol Garfunkel, Executive Director of COMAP, which had been supporting applicable mathematics in a variety of ways since 1972. Garfunkel was very enthusiastic, and Page urged me to get in touch with him.

Although I was a member of COMAP and had used its materials, I had never had any interaction with Garfunkel. After one phone conversation, it was clear that we had similar goals. In fact, my personal campaign to "increase the applied mathematics presence on campus" might be one way to describe what COMAP had been doing over the years. He suggested that a proposal for a three-year grant be sent to the Fund for the Improvement of Post-Secondary Education (FIPSE) of the U.S. Dept. of Education, with COMAP the administering body and me as the Project Director. FIPSE had a reputation for backing novel ideas that might have far-reaching effects. The derivative term "Applied Putnam" was transformed into "Mathematical Competition [now Contest] in Modeling." A preliminary proposal made FIPSE's January 1984 deadline, and the three-year proposal was approved in June 1984.

Goals

The goals and purposes of the MCM are best described by two paragraphs from the abstract of the proposal to FIPSE:

> The purpose of this competition is to involve students and faculty in clarifying, analyzing, and proposing solutions to open-ended problems. We propose a structure which will encourage widespread participation and emphasize the entire modeling process. Major features include:
>
> - The selection of realistic open-ended problems chosen with the advice of working mathematicians in industry and government.
> - An extended period of time for teams to prepare solution papers within a clearly defined format.
> - The ability of participants to draw on outside resources including computers and texts.
> - An emphasis on clarity of exposition in determining final awards with the best papers published in professional mathematics journals.
>
> As the contest becomes established in the mathematics community, new courses, workshops, and seminars will be developed to help students and faculty gain increased experience with mathematical modeling.

Organizing

Garfunkel and I were in firm agreement that the contest must be primarily an educational experience, not a competitive one. In a sense, we wanted it to be closer to the spirit of traditional English sport than to modern American sports.

I formed an advisory board of mathematical scientists who had been early backers of an applied mathematics contest:

- A.P. Hillman, University of New Mexico;
- M.S. Keener, Oklahoma State University;
- H.O. Pollak, Bellcore;
- F.J. Roberts, Rutgers University;
- E.Y. Rodin, Washington University;
- L.H. Seitelman, Pratt & Whitney;
- Maynard Thompson, Indiana University; and
- myself as chair.

Hillman, who for many years directed grading for the Putnam, agreed to be chief grader. This coup eliminated one of the two swords that hung over our heads: finding suitable problems, and judging.

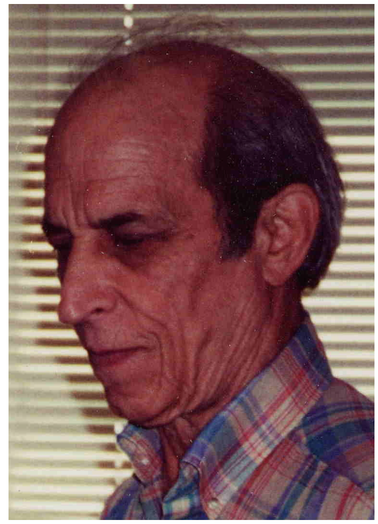
The advisory board first met in August 1984. We selected two types of problems, approved ground rules, set up a Putnam-like system of faculty advisors, and established a classification for solution papers. We set the inaugural contest for the weekend of 15 February 1985. The meeting was very productive; but we departed with a note of concern over the short amount of time for publicity, registration, and final write-up of contest materials. We wondered whether we could get our predicted 55 colleges to enter the first contest.

It turned out that 158 teams, representing 104 colleges, registered for the first contest, a response that overwhelmed us. Any more than 100 solution papers would be unmanageable; there wouldn't be enough judges to allow for multiple readers for each paper. It turned out that 90 papers, representing 70 colleges, were submitted—a large but tractable number. The MCM was a success!

Acknowledgment

This history of the foundation of the MCM is adapted from the author's "Mathematical Competition in Modeling" in *Mathematical Modelling* [continued as *Mathematical and Computer Modelling*] 6 (6) (1985): 473–484.

About the Author



Ben was the founder of the MCM and its director for the first seven years. He has a B.A. from Swarthmore College, an M.A. from Columbia University (analysis), a Ph.D. from the University of Maryland (partial differential equations), and most recently (1990) an M.A. from the University of Maryland (computer science).

He taught at several other colleges and universities before going to Salisbury State in 1974, where he served as chair of the Mathematics and Computer Science Dept. 1974–82 and received the Distinguished Faculty Award in 1992. Ben was NSF Lecturer at New Mexico Highlands University and at the University of Oklahoma, Fulbright Professor at National Taiwan Normal University, and visiting professor at the U.S. Military Academy at West Point. He has taught most undergraduate mathematics courses, plus graduate courses in integral equations, partial differential equations, and mathematical modeling.

In recent years, Ben has been a major exponent of environmental mathematics, a topic on which he has presented several minicourses.

Ground Rules for the MCM

Registration of Teams

The team advisor must register the team(s) in advance using the contest Website (detailed instructions are at the URL cited at the end of this chapter). Each department may sign up either one or two teams of up to three undergraduates each, but no more than three teams may be registered from any one institution. Team members may be changed at the last minute without notifying COMAP. If a department preregisters only one team but later wishes to add a second team, the department must obtain a new control number for the second team.

Contest Date and Time

The contest is held on a weekend in February. In recent years it has begun at 8 P.M. EST local time on a Thursday and ended the following Monday at 8:00 P.M. EST.

The team advisor must send to COMAP three copies of the solution paper to arrive by a specified deadline. The paper must be typed and in English and include paper copies of any supplemental diagrams, graphs, computer programs, etc. No non-paper supplementary materials (diskettes, videotapes, etc.) are accepted.

The Advisor

The advisor is the key to the success of MCM. The advisor alerts students to the competition and its benefits, encourages the organization of teams, and registers the teams for the contest. It is both legitimate and desirable to coach and prepare teams.

Rules

Each team receives the same two problems to consider but may submit a solution paper for only one.

Teams may use any inanimate source of data or materials—computers, software, references, web sites, books, etc., however all sources used must be credited. Failure to credit a source will result in a team being disqualified from the competition.

Team members may not seek help from or discuss the problem with their advisor or anyone else, except other members of the team. Input of any form from anyone other than student team members is strictly forbidden. This includes email, telephone contact, personal conversation, communication via web chat or other question-answer systems, or any other form of communication.

Neither the school name nor team members' names may appear anywhere in the solution paper; they may appear only on the control sheet.

Suggested Outline

- A clarification or restatement of the problem, as appropriate.
- A clear exposition of all variables, parameters, assumptions and hypotheses.
- An analysis of the problem motivating and justifying the modeling to be used.
- The design of the model.
- A discussion of how the model could be tested, including error analysis and stability (conditioning, sensitivity, etc.).
- A discussion of the apparent strengths or weaknesses of the model or approach.
- A one-page summary of the results, typed on the Summary Sheet, must be attached to the front of each copy of the solution paper.

Judging

Partial solutions are acceptable. There is no passing or failing cut-off score, nor will numerical scores be assigned. The judges are primarily interested in the team's approach and methods. Conciseness and organization are extremely important. Key statements should present major ideas and results.

Each solution paper must include a one-page typed summary on the Summary Sheet included with the contest materials. *It is unlikely that MCM judges will read beyond a poorly constructed summary.* The summary is a very important part of the MCM paper. The judges place considerable weight on the summary, and winning papers are sometimes distinguished from other papers based on the quality of the summary. A summary should clearly describe the approach to the problem and, most prominently, the most important conclusions. The concise presentation of the summary should inspire a reader to learn the details of the work. Summaries that are mere restatements of the contest problem, or are a cut-and-paste boilerplate from the Introduction, are generally considered to be weak.

Results, Recognition, and Prizes

Judging takes place a few weeks after the contest. Solutions are recognized as Successful Participation, Honorable Mention, Meritorious, or Outstanding.

The advisors and teams are notified of the results in April. COMAP issues news releases and announcements in college and professional publications.

All successful participants receive a certificate. Outstanding teams receive bronze plaques, and each year their solution papers have been published in a special issue of *The UMAP Journal*.

INFORMS, the Institute for Operations Research and the Management Sciences, designates one Outstanding team from each problem as an INFORMS Winner. Each member of these teams receives a three-year membership in INFORMS and a cash prize. Each member of the other Outstanding teams receives a one-year membership in INFORMS.

The Society for Industrial and Applied Mathematics (SIAM) designates one Outstanding team from each problem as a SIAM Winner. Each designated team receives a certificate, a cash prize, and partial expenses for a trip to the SIAM annual meeting.

The Ben Fusaro Award honoring the Founding Director of the MCM was awarded for the first time in 2004. It recognizes one of the Meritorious or Outstanding teams for an especially creative approach to the contest problem.

Details of the procedures for an upcoming contest, plus answers to frequently asked questions, are available at

`http://www.comap.com/undergraduate/contests/mcm/instructions.php` .

Major funding and support for the MCM is provided by the U.S. National Security Agency and by SIAM. However, to cover remaining costs, COMAP must charge a per-team registration fee, to be paid by credit card.

Judging the MCM

Frank Giordano, MCM Director
Naval Postgraduate School
1 University Circle
Monterey, CA 93943-5000
frgiorda@nps.navy.mil

Each paper submitted is classified as Non-successful, Successful Participant, Honorable Mention, Meritorious, or Outstanding. Further, from among the Outstanding papers, judges from SIAM, INFORMS, and the MAA pick winners to receive awards from their societies. Typically, the top 2–3% are classified as Outstanding, the next 15% as Meritorious, and the next 25% as Honorable Mention. A paper classified as Successful Participant is a complete paper that satisfies the rules of the contest.

The process of determining the classification of each paper consists of distinct phases: screening, further screening, and judging. Each phase employs a mixture of normalized grading, ranking, classification against a judge's "absolute ideal," and qualitative judging. We describe each of the phases.

Screening Rounds 1 and 2

There are three screening rounds, to identify papers that are not going to be among the top 43% (Honorable Mention or above). After a preliminary reading of several papers (not graded), judges jointly design a 7-point grading scale. Discriminators used on the scale are designed to classify the papers into three categories. The top classification reflects exceptional quality; the second category are papers that the judge feels should be retained; and the third category are papers that the judge thinks should be eliminated. The judging scale emphasizes heavily the organization of the paper: What ideas have the contestants used in developing and analyzing their models? The summary is weighted heavily, as the contest rules require that it reflect the major ideas that the contestants used. Typically, a judge devotes 10–15 minutes to screen a paper.

The first two of the three screening rounds are accomplished by "Triage Judges." Each paper is read by one judge in each round. After the second round, papers judged in the lowest category by both the judges who read them are eliminated. If both judges vote the paper weak, but their scores differ by more than 2 points, a third judge classifies the paper before it is eliminated. Typically, about 25% of the papers are eliminated after the first two screening rounds.

Screening Round 3

Further judging takes place under a different set of judges. Normalized scores from the first two screening rounds are used to rank-order the papers. The normalized scores are used to organize the papers into as many stacks as there are judges. Upon arrival, judges are given a stratified packet of papers to read for "calibration" (not grade). After reading the papers, the judges jointly design a 7-point screening scale. The judges then conduct a third and final screening round.

After the screening round, the judges design a 100-point scale for the final judging.

Final Judging

About 43% of papers advance to final judging. Typically, there are four final judging rounds. A judge typically spends 30–45 minutes to judge each paper. In addition to judging the paper, the judge is asked to rank it against all papers read in the round. The judge is also required to assign an "absolute" classification: Successful Participant, Honorable Mention, Meritorious, or Outstanding. This classification permits the judges to render an opinion on the "absolute" quality of the paper independent of grading and ranking procedures.

After the three screening rounds and the first two final judging rounds, about 18% of papers remain. After two more final judging rounds, typically 6–12 papers remain. Time is provided for judges to read papers that they have not yet read. The judges then meet to debate and compare the merits of the papers. The Outstanding papers are chosen by consensus.

Stratified Packets

Beginning with the third screening round, the papers are organized into as many stacks as there are judges. The papers are distributed modulo the number of judges, using the cumulative normalized scores. The cumulative scores are weighted, with the final judging rounds counting more than the screening rounds.

Eliminating Papers

Typically, about 25% of the papers are eliminated after the first two screening rounds. Beginning with the third screening round, the Contest Director, the Associate Contest Director, and the two Head Judges meet to discuss the elimination of papers. The information that they use is

- the overall rank based on normalized scores,
- the round rank assigned each round, and
- the overall classification given to the paper by the judge.

Each item is quite useful in determining which papers should be eliminated. Since judges receive a stratified packet, the round ranks are especially useful. No "quota" is used to determine how many papers to cut each round; however, typically the top 43% (Honorable Mention, Meritorious, and Outstanding) remain after the first screening round, and the top 18% (Meritorious and Outstanding) remain after the third grading round.

The Judges

The judges are led by experienced Head Judges who graded for several years before becoming a Head Triage Judge. The professional societies choose their own judges from among volunteers with strong credentials. COMAP rounds out the judging team by choosing a field of judges with a wide range of expertise. If needed, the field of judges is augmented with subject-matter experts.

Conclusion

Judging a contest that receives as many creative student solutions as the MCM does is a difficult task. The judges deserve our gratitude for a difficult job done well, and with great dedication and integrity.

About the Author

Frank Giordano has been Director of the MCM since 1991. He is a Distinguished Honor Graduate of the U.S. Military Academy and has an M.S. in Management Science and a Ph.D. in Operations Research from the University of Arkansas. In addition, he was a scholar on the George Olmsted Foundation at the University of Madrid, where he received various certificates from the College of Philosophy and Letters and the equivalent of an M.S. degree from the College of Civil Engineering. From 1975 to 1995, he was a professor of Mathematical Sciences at the U.S. Military Academy, rising to department head and vice-dean; he retired from the U.S. Army in 1995 as a brigadier general. He spent 1995–96 as E.L. Wiegand Distinguished Visiting Professor at Carroll College (Montana), and since 2002 he has been Professor of Defense Analysis and Operations Research at the Naval Postgraduate School. Over the past decade, he has prepared and directed various NSF grants for COMAP.

Ability, Training, Preparation

Anne M. Dougherty
Dept. of Applied Mathematics
University of Colorado
526 UCB
Boulder, CO 80309–0526
anne.dougherty@colorado.edu

Introduction

Since 1997, Bengt Fornberg and I have been advisors for Modeling Contest teams at the University of Colorado at Boulder. Through 2005, we have worked with 23 teams and have had consistent success: 6 Outstanding, 9 Meritorious, 7 Honorable Mention, and 1 Successful Participant designations.

Over the past few years, I have given considerable thought to the ingredients that make an Outstanding team. *I believe that success in the MCM comes down to equal parts student ability, training, and team preparation.* Take bright and talented students, give them challenging coursework that includes mathematical and computational work, prepare them in teams of three for the MCM, and then let them go. The results are exciting.

Students

The University of Colorado at Boulder has some of the most talented undergraduates anywhere. After graduating from high school, these students are looking for additional challenges—although they don't always know what that means. They are bright and articulate, and our goal is to encourage each of them to develop fully. These students have different majors in science, engineering and applied mathematics; but as they pass through our calculus and differential equations courses, we encourage them to "take more math courses." Some will become applied math majors, double majors, or minors; but all will benefit from additional mathematical and computational training.

Training

The mantra of the Department of Applied Mathematics is "mathematics, computation, and communication." Each of these skills is infused throughout our curriculum. As early as Calculus 3, our students are required to work on computer projects using Mathematica or Matlab. Each major, and most of our minors, takes a minimum of one programming course and one course in numerical analysis. Computer software is used extensively throughout our upper-division courses.

Our best undergraduates, many of whom are MCM participants, are also encouraged to work on research projects with CU faculty. Students have worked in physics labs, on significant programming projects, and on research projects with applied math faculty. While working on these projects, students don't have the time pressure of the MCM. Nonetheless, they do work on open-ended research problems, grappling with some of the same issues as those posed in the contest problems. Through coursework and research projects, our students receive extensive mentoring and training in skills that will serve them well in both the MCM and in future graduate work or careers.

Team Recruitment and Preparation

The primary role of the advisor in the MCM is team recruitment and preparation. Students are recruited from throughout the science, engineering, math, and applied math disciplines. The best teams are those where each student brings skills that complement those of their teammates. We recruit students from our classes, via email, and by word of mouth spread through students. We always stress that students must want to do the contest—they need to feel excited and challenged. As soon as possible, even as early as the fall semester, we encourage students to form their teams. Thus, the individual excitement that they feel is transferred to team enthusiasm.

The MCM is a unique event that students must be prepared for. You can have students with enormous amounts of ability and good training; but if they are not prepared for the nature of the contest, they may not do as well as possible. To prepare students, we have one or two meetings before the winter break and then three meetings afterwards. After going over the contest rules, we discuss division of labor, timing, and the nature of the final paper.

It is important that each team understand that a division of labor is critical. While the character of each team is slightly different (and depends on the problem), there are usually three roles: *the writer, the programmer, and the researcher*. After some initial brainstorming and research by all team members in the first few hours of the contest, the team should have some ideas for their first, and most simplistic, model. At this point, the writer can begin writing the introduction, the programmer begins the numerical work for the first model, and the third member continues the mathematical development and library (or Internet) research.

Beginning the writing on Friday cannot be overstressed. One of my first teams was exceptionally talented, but they did not understand (and I had not stressed) the importance of this timing. Each person worked on the research, model development, and programming. They actually developed several models, from the simple to the very sophisticated. Unfortunately, only the simple model made it into their paper. They didn't start writing until Sunday night—

and then all three wrote furiously. The resulting paper was an unedited jumble of ideas, a real disappointment. I now stress, multiple times during our preparations, how important it is for one person to be responsible for the paper—and that he / she needs to start writing early.

During our practice sessions, we also discuss the format of the paper. The problem restatement, the assumptions, the models, and the conclusion are all examined. We also talk about the word-processing itself. Some teams prefer to work in LaTeX; others prefer Word. I tell students that they should use whichever they feel most comfortable as long as they can easily incorporate equations, graphs, and tables. I do encourage students to set up a template before the contest begins and to make sure that they can import graphs and figures.

The last topic that we discuss is where each team will work. Each team needs its own separate workspace. Some teams prefer to work at a team member's home or apartment. Others prefer to work in one of the computer labs. We also provide weekend access for one team to the departmental conference room.

Aftermath

After the contest results are announced, we make sure to publicize them throughout the university. In April, we host the Modeling Contest presentations. Each team is given 15–20 minutes to present their solution. All undergraduates and faculty are invited to attend. It is great for each team to be able to share their solution with everyone else. Students who did not participate in the contest are inspired to do it the following year. It is always amazing to me how much these students can accomplish in just 96 hours. Their energy and enthusiasm motivate me. I believe that these students, who have accepted the MCM challenge and given it their best, will remember this event for the rest of their lives.

About the Authors

In 1994, Anne Dougherty received her Ph.D. in probability from the the University of Wisconsin–Madison. She then joined the Dept. of Applied Mathematics at the University of Colorado at Boulder as an instructor. She is currently a senior instructor and Associate Chair of the department. Her research interests include applied probability and statistics and image analysis. She has been working with and advising MCM teams since 1997.

Bengt Fornberg received his Ph.D. in numerical analysis from Uppsala University in Sweden in 1972. He held several university positions from 1972 until 1984. He then worked for Exxon for 11 years before joining the Dept. of Applied Mathematics at the University of Colorado at Boulder in 1995. His research interests include computational fluid dynamics, pseudospectral methods, and industrial applied mathematics. He has been working with MCM teams since 1997.

Making Math Exciting

James Allen Morrow
Dept. of Mathematics
University of Washington
Box 354350
Seattle, WA 98195-4350
morrow@math.washington.edu

The Mathematical Contest in Modeling is an important event at the University of Washington. It has attracted our best students, and the success of our teams has made the Department, the University, and the community very proud. Publications such as the *University of Washington Daily*, *Columns Magazine*, the *Seattle Times*, and the *Seattle PI* have published articles about our winning teams. The *Seattle Times* even wrote an editorial praising them. The winners have been introduced at special events and praised highly by the UW President at Regents' meetings. The result is that the Mathematics Department has become widely respected, and consequently outstanding students are selecting mathematics as their major.

Publicity about our success has contributed to continuing interest in the contest, making it easy to recruit talented and enthusiastic participants. We are usually able to form some teams consisting of experienced competitors and others with promising newcomers. It is important to give younger students the chance to participate, since experience in the contest is a good (but not perfect) predictor for success. The contest is also addictive—students are anxious to compete again and are willing to spend time preparing.

Teams are formed in October so that they will have several months to work together. It is a good idea to have a variety of talents on each team. Many team members are double majors and have experience using mathematics to solve problems. Interest is such that we are also entering teams in the ICM.

Preparation for the contest consists of

- frequent team meetings;
- reading winning papers;
- discussions of judges' comments;
- discussions with previous teams (particularly winning teams), who talk about their experiences and give advice;
- preparation of facilities (office space, computing accounts and equipment, library access, printers), and
- arranging the working environment well in advance.

Students need to become familiar with

- LaTeX, in order to minimize time dealing with document preparation—this means making sure that appropriate software is available and correctly installed; and
- library and on-line resources that will have to be used to find information.

Research experience and experience working on a team are very important. The difference between classwork and tests is enormous. Students need to have the experience of formulating problems and dealing with inevitable failed attempts to find solutions. A good way to gain this experience is to participate in summer Research Experiences for Undergraduates (REUs), work on independent projects, and work with a group to attack problems in specific areas.

Students need to learn to work together to contribute ideas and constructive criticism. We try to form teams that have compatible personalities; in many cases, the teams consist of students who have already been working together informally on homework and test preparation. Each team will have its unique way of assigning responsibilities. One person may have computational skills, one person may have mathematical modeling experience, another may have good expository skills.

An intangible yet definite feature of the contest is the excitement that it generates. It has become a high point of the year for the students. It is their Super Bowl and they await it nervously, yet with confidence. The energy of the teams seems boundless. Only someone with the physical strength of youth can exert such an intense, concentrated, effort. Our faculty marvel at the quality of the output produced in such a short time under stressful circumstances. The results have been so rewarding that the University and the wider community eagerly anticipate the event and constantly ask for information.

The best way to end this essay is to quote some of the students.

- An MCM contestant: “I am glad (after 15 hours of sleep, that is) that I did this contest. I got to work with such brilliant minds on a problem that I would have never done otherwise. We learned so much in the process and got pushed so hard. Now every time there is a stressful situation, I’ll always think back about the MCM and realize that there could be worse situations. It was an unparalleled experience all together.”
- An ICM contestant: “At the start of the contest, we said that our first and foremost goal was to be better friends at the end than the beginning, and I’m very proud that we succeeded. We were able to come to consensus on almost everything, and work out our difficulties when we couldn’t.”

Perhaps COMAP didn’t realize when the contest was first suggested how much it would do to make math exciting!

About the Author

James Morrow graduated from Adamson High School in Dallas TX in 1959. He received a B.S. in mathematics from Caltech in 1963 and a Ph.D. in mathematics from Stanford in 1967; his thesis advisor was Kunihiko Kodaira. He was an Instructor at the University of California–Berkeley, from 1967 to 1969, and he has been at the University of Washington since 1969, where he is Professor of Mathematics. He served one term as Graduate Program Coordinator, two terms as Undergraduate Program Coordinator, has directed an REU program since 1988, and has been the director of Mathday for high school students since 1993. He has been the advisor of four Outstanding MCM teams. He won the Distinguished Teaching Award at the University of Washington in 2003.

Go with What You Know

W. Garrett Mitchener
Department of Mathematics
Duke University
P.O Box 90320
Durham NC 27708
wgm@math.duke.edu

A problem doesn't have to be fancy to be difficult.
—Michael Reed

The Mathematical Contest in Modeling is very different from most math contests. Having participated as a student and advised teams as a faculty member, I offer the following suggestions about what skills should be present in an MCM team. My main point is:

Good teams have a mixture of mathematical, computational, and writing skills.

I suggest that one member become the team's specialist in each of these areas, though all members usually contribute skills to all three. Unlike other contests, in the MCM problem-solving tricks are not as useful. Instead, teams must

- make creative use of what talents they have,
- know how to best use them in the limited time of the contest, and
- know how make good judgments about what tools to use and learn in the limited time of the contest.

What Mathematics to Use

I find that the most useful mathematical areas are probability and dynamical systems, because almost every modeling problem involves uncertainty, variability, and changing quantities. Specific problems may require knowledge from other areas. For example, graph theory might be useful for problems concerning networks, PDEs are often appropriate for materials-science problems, and an image-analysis problem may require signal processing. Obviously, no one team member should plan to be an expert in all things mathematical, nor should anyone plan to become an expert in PDEs in one weekend.

Some research will no doubt be necessary, and if you decide to read books or articles to learn a specialized topic, do the necessary reading but know how to stop before reading becomes a time sink. With that in mind, perhaps *the most useful mathematical skill is the ability to improvise*: You should be as creative as possible with what you *do* know. Improvising can improve your paper by giving the problem a fresh perspective that will not be found through research. Additionally, the problems generally don't come with enough information for you to solve them completely. In real life, you would ask the problem poser for the information, but it may not be available, and it will certainly not be available during the contest, so very often you must improvise. For example, you might make additional assumptions that allow you to simulate the missing information. From there, you can show what you would do with the actual information if it were available.

Use what you know, but improvise as needed.

Computation

Computation and simulation are indispensable tools for applied mathematics. Someone on the team should be skilled in software development, and know how to

- write good programs rapidly;
- estimate the total time required to write, debug, run, and analyze a proposed computation; and
- propose alternatives to computations that will take too much time.

I prefer programming languages such as Python that are high-level and easy to use. A Python program that takes two hours to develop and runs in fifteen minutes is often better than an equivalent C++ program that runs in seconds but takes two days to write and debug. I remember several situations where I wrote a program in C++ and spend tremendous amounts of time fixing memory bugs and compiler issues. In the end, despite the speed of C++ the program turned out to be so time-consuming that it never finished running. With 20/20 hindsight I can say: There simply isn't enough time in a weekend for that. You don't want to end up with a paper that just says you designed a simulation but couldn't run it, with no results or analysis.

Mathematica, Maple, and Matlab can be indispensible time-savers, since they include high-level programming languages with built-in solving and graphics commands. However, if you plan to use one of these, start learning it well ahead of time; you don't want to waste a lot of time during the contest trying to learn a complex software package. As with mathematics, I recommend above all that you make the most of what you already know. If you discover during the contest that there are better tools and methods available, see if you have time to learn them, but consider leaving them for next year.

Program in what you know, but in advance learn a useful language well.

Writing

Writing skills at all levels of detail are crucial. It's surprisingly easy to spend all weekend on a problem and write an incomprehensible paper—without realizing it. You understand the problem and your team's solution thoroughly (well, you hope so); but to write a good paper, you must step back and adopt the perspective of someone who knows nothing about what you did.

The paper needs to have consistent notation, a list of references, and appropriate graphics. It should be organized, coherent, and readable. Identify vague points in the paper. For example, if you find yourself writing that a certain result "looks better," figure out specifically what makes it look better and use that as an opportunity to improve the analysis of your result and make a stronger, more correct, and more precise statement in your paper.

The judges are looking for specific features, such as

- clearly-stated assumptions,
- a thorough understanding of the problem (including natural questions that weren't specifically asked in the problem statement), and
- a clear description and analysis of your model.

Someone on the team needs to be in charge of making sure that all of these features are present and easy for the judges to find. Given that the task of writing will often be split among all three team members, someone should be in charge of assembling fragments written by each member and checking them for consistency and redundancy. Most models come in phases of increasing complexity, and it's crucial that the paper clearly state which assumptions and conclusions are true of which phases. With three contributors, confusion among phases and inconsistent notation can easily creep in. Someone must be responsible for finding and eliminating all these potential sources of errors so that the explanation of the model and its behavior are clear.

On a more mechanical level, someone needs to be proficient with the typesetting or writing software you use, such as LaTeX or Word, so when you need to add page numbers, merge revisions, or import graphics in a particular way, you don't waste time looking through manuals or online help. Again, go with what you know; and if you don't know how to do a particular task, spend a reasonable amount of time trying to figure it out, but know when to improvise and go with Plan B.

Write what you know, as if your reader doesn't know about it.

Conclusion

In conclusion, the MCM challenges teams to balance creativity against existing knowledge, and quantity against quality. Team members should be aware of their own strengths and weaknesses, and make certain that mathematical, computational, and expository skills are all covered. They should be prepared to explore new tools during the contest, but with the realization that it's often best to go with what you already know.

About the Author

I'm a graduate of the North Carolina School of Science and Mathematics (NCSSM) and Duke University, and I participated in the MCM at both schools. Once at NCSSM and twice at Duke, my teams earned the Outstanding designation. I earned my Ph.D. at Princeton University in applied and computational mathematics, and I have returned to Duke as a post-doctoral research associate. Currently, I help organize Duke's MCM teams. My research is in mathematical modeling for linguistics. In my spare time, I play oboe for the Durham Community Concert Band, dabble in gardening and calligraphy, and lately I've been learning to juggle. I live in Durham NC and attend Blacknall Presbyterian Church.

Honor Roll

Edward E. Allen
Hugh N. Howards
Miaohua Jiang
Stephen B. Robinson
Dept. of Mathematics
Wake Forest University
Winston-Salem, NC 27109
allene@wfu.edu

At Wake Forest, we have had good success in the Mathematical Contest in Modeling. In the years 2000–2003, five of our teams received the "Outstanding" designation, including two teams in 2002. Also, those teams received each of the major prizes in that period: SIAM (twice), MAA, and INFORMS.

There are several reasons that we believe we have had success.

- **Make it fun.** We make the competition as much fun and as rewarding as possible for the participants. We announce the contestants on the teams. The competition is a good project for students. We write letters of recommendation that include their participation in this activity. We also award one semester hour of Pass / Fail credit for participation. We have many students who participate in the competition a second or third time.

- **Maintain focus.** We help the students focus on the competition. It is very important that the participants do not get frustrated during the contest. To make their participation easier, we give the teams a budget for food. We find rooms for them to use for the whole competition. Faculty have been known to give up their offices to a team for the entire length of the competition if no other suitable space can be found.

- **Recruit.** We recruit aggressively. It is hard to get students' attention. Just because an activity is fun and worthwhile does not mean that students will automatically participate. It is not difficult, however, to find students to participate. One year, we only had one team ready but wanted two teams to participate. One advisor twisted the arms of his best three students in a single course. It took a bit of cajoling, but that team of two women and one man took an Outstanding designation and the INFORMS prize.

- **Scout out talent.** We look for more than just mathematics majors. While mathematics majors usually have the best feeling for the mathematics, individuals in other majors may have wonderful ideas about modeling that complement the skills of the mathematics majors. The departments of physics, computer science, and English are good sources for participants.

- **Share the work.** We share the responsibility as advisors. We have many faculty who are willing to sponsor a team, so we do not all get to advise a team every year. Although a team may have one official coach, several different faculty members will meet with the teams to share ideas and experiences. Other departments, besides mathematics, are also willing to sponsor teams.
- **Stress writing.** We emphasize the importance of good exposition. Wake Forest requires as part of its graduation requirements that its students take many more courses in the humanities than most universities. Our students, even in the sciences, are very good writers. In preparation, we emphasize that excellent writing is absolutely required. Long or hard-to-understand proofs either need to be referenced or put into an appendix.
- **Have a common computer culture.** Our students have a common computer platform. All undergraduates at Wake Forest receive a laptop computer. For the contest, having a single platform helps communication and the transfer of information and makes the writing process much easier.
- **Prepare.** We review old contest problems and solutions, to help the students understand what types of solutions are required. Veteran participants are good sources of information for newer participants on how to pace a team during the competition.
- **Practice.** We ensure that the students have practiced writing up their solution in whatever format they are going to use. If the students are going to use TeX or MS Word, we make sure that they are familiar with that system before the competition starts.
- **Celebrate!** Students at Wake Forest have developed a tradition of "rolling the quad" after significant wins in athletic competitions (see **Figures 1–3**), a tradition known elsewhere as "TP-ing" (from the initials of "toilet-papering").

Figure 1. Space.

As advisors, we decided that our modeling teams deserved no less. After taking an Outstanding, we get the Mathematics Club to use approximately 200 rolls of toilet paper to decorate the quad in honor of our teams.

Figure 2. Operator.

We invite the President of the University and the Dean of the College to participate, and the university photographer and the local TV news stations to observe. There is a certain excitement on the quad when students and the faculty from the Mathematics Dept. celebrate by throwing rolls of toilet paper over the trees. Faculty from other departments have mentioned how much they appreciate that we celebrate academic accomplishments as much as athletic wins.

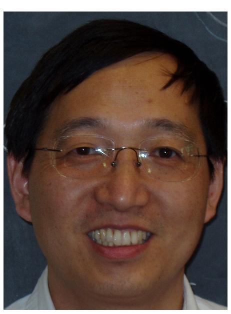
Figure 3. Transformed space.

The combination of technical, communication and creative skills that this competition requires is a good fit for our institution. We owe special thanks to the organizers and judges of this competition. Finally, we would like to believe that the key to our success in this competition is the excellent collection of advisors; but the fact of the matter is that it is the creative, hardworking, and talented students who make this competition such a wonderful experience at Wake Forest.

About the Authors

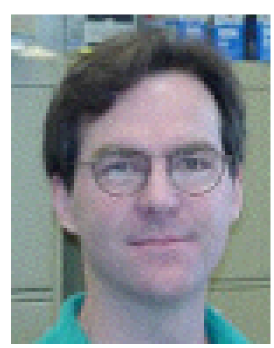
Edward E. Allen is a Professor of Mathematics at Wake Forest University. He received his B.S. in Mathematics from Brigham Young University and his Ph.D. from the University of California, San Diego. His research is in the areas of algebraic combinatorics and bioinformatics. He is a co-principal investigator on a grant from the National Institute of Health that explores the modeling of signal transduction networks in cancer cells.

Hugh N. Howards (shown with WFU's MCM award plaques) is a Sterge Faculty Fellow and Associate Professor of Mathematics at Wake Forest University. He received his B.A. in Mathematics from Williams College and his Ph.D. from the University of California, San Diego. His research is in low dimensional topology. He won the campus-wide Reid-Doyle Prize for the top young teacher at Wake Forest in 2004.



Miaohua Jiang is an Associate Professor of Mathematics at Wake Forest University. He received his B.S. in mathematics from Wuhan University, M.S. in mathematics from East China Normal University, and Ph.D. in mathematics from Pennsylvania State University. His research is in the areas of dynamical systems and ergodic theory.

Stephen B. Robinson is a Professor of Mathematics at Wake Forest University. He received his B.A. and Ph.D. in Mathematics from the University of California, Santa Cruz. His research is in the areas of nonlinear analysis and the geometry of medical imaging.



20 Years of Good Advice

Pat Driscoll
Department of Systems Engineering
U.S. Military Academy
West Point, NY 10996

Jerrold R. Griggs
Dept. of Mathematics
University of South Carolina
Columbia, SC 29208

Mark Parker
Dept. of Mathematics, Engineering, and Computer Science
Carroll College
Helena, MT 59625

Paul Boisen
National Security Agency
Ft. Meade, MD 20755

William P. Fox
Dept. of Mathematics
Francis Marion University
Florence, SC 29501

Michael Tortorella
Dept. of Industrial and Systems Engineering
Rutgers University
Piscataway, NJ

Paul J. Campbell
Beloit College
Beloit, WI 53511

The following advice is synthesized and distilled from commentaries by judges of the MCM and ICM over the years.

The Model Should Be Based on Research . . .

Teams are increasingly adept at using the Internet to find credible information sources to support their modeling efforts, but there is a good deal of room for improvement in how to incorporate such information into their papers, especially for a team that perceives that it has struck the mother lode of reference sources. Incorporating others' work without diluting one's own effort is challenging. Parroting large portions of technical reports, thereby reducing the team's contribution to simply interpreting someone else's research, is not the solution.

Three uses of existing research are common in technical reports:

- To chronicle the events leading to the approach taken in the current paper and to help the reader understand the context or domain of the problem. This action is typically accomplished in an Introduction or Background section.
- To identify and justify technical parameters needed for the new approach.
- To compare the graphical, symbolic, or numerical results generated by the new modeling approach with those previously identified, so as to examine the benefits or drawbacks of the new approach.

Credible existing research used in these ways does not replace or dilute the current effort but directly supports and strengthens it.

The judges look for evidence whether a team actually did some modeling of its own rather than simply looking up a few equations and trying to shoehorn those into the problem. Experiments can be good to see, too, if appropriate.

Given the time pressure of the contest, a team has to be cautious not to get trapped into adopting a complicated modeling component from existing research without being able to explain clearly its development, its use and limitations, and its impact on the current model. This temptation is the classic red herring of the contest, luring teams into committing to an approach—only to discover late in the process that they are ill-equipped to handle it. Ultimately, evidence of this error appears in the paper as miraculously appearing formulae, unexplained graphics, and tables of data still waiting to be analyzed. Just as in a court of law, the judges consistently find the results of models built on such tenuous foundations difficult to believe.

. . . Must Produce Results . . .

Develop your model—do not just provide a laundry list of possible models. Start with a simple model and then refine it. Also, it is far better to work out one model thoroughly than to present several half-baked approaches. Judges are definitely not interested in a blow-by-blow historical narrative of what you tried that didn't work.

Some papers have noticeable gaps that call into question the validity, veracity, credibility, and applicability of the results presented. Consequently, if a team's principal effort is, say, to construct computer code to simulate an air traffic scenario, *they must present evidence that their code/model actually ran and yielded the information sought*. Analyzing the output of a model provides a basis for determining if the modeling approach chosen was reasonable.

. . . Which Must Be Analyzed . . .

Simply creating an acceptable mathematical representation (system of equations, simulation, differential equations, etc.) of a real-world event is not enough. The representation (model) must be tested to verify that the information that it produces (solutions, simulation output, graphics, etc.) makes sense in the context of the questions asked and the assumptions made. *It is insufficient to present such a representation without this additional evidence.* Once a mathematical model is created, use

- symbolic,
- graphical, and / or
- numerical

methods to exhibit evidence that the model works. Many of the best papers use a combination of these three approaches; some teams write computer code or use spreadsheets, while others use a computer algebra system as their workbench.

. . . and Compared with the Assumptions

Papers reaching the final round of judging paid attention to

- stating their assumptions clearly,
- avoiding making assumptions that are never used or not really needed,
- explaining the impact of each assumption, and
- telling why they felt it was necessary to include it in their model development.

They were also careful not to assume away the challenging and information-relevant portions of the problem posed. It is easy to follow the logical construction of these teams' models and to identify what they were attempting to do. However, sometimes even a very good paper mistakenly places key information in appendices rather than in the section where supporting evidence is desperately needed.

Crucial Elements in an Outstanding Entry

A thorough, informative summary is essential. Your summary is a key component of the paper; it needs to be clear and contain results. It should not say, "Read inside for results." The summary should motivate a judge to read the paper to see how you obtained your results. Even a paper that is otherwise strong can be eliminated in early judging rounds because of a weak summary. Many teams mistakenly form the summary by simply copying parts of the introduction of the paper, which has the different purpose of establishing the background for the problem. On the other hand, the summary should not be overly technical. A long list of techniques can obscure your results; it is better to provide only a quick overview of your approach. Don't merely restate the problem, but indicate how it is being modeled and what was learned from the model. Put into the summary the "bottom-line and managerial recommendation" results—*not a chronological description of what you did.*

Develop a model that people can understand. The model should be easy to follow. While an occasional "snow job" may make it to later rounds of judging, we generally abhor a morass of variables and equations that can't be fathomed. Well-chosen examples enhance the readability of a paper. It is best to work the reader through any algorithm that is presented; too often papers include only computer code or pseudocode for an algorithm without sufficient explanation of why and how it works or what it is supposed to accomplish.

Supporting information is important.

- Figures, tables, and illustrations can help demonstrate ideas, results, and conclusions and thus help sell your model, but *you must refer to these aids in the text of the paper and explain them*. Each such display should have a caption that tells what is in the display, and the display should indicate the measurement units of quantities. Graphs should have scales and axis labels.
- A complete list of references is essential—document where your ideas come from.

Follow the instructions.

- Answer all the required parts and make it clear that you have done so. Attempt to address all major issues in the problem. What the judges pay attention to is whether or not the team engaged the questions asked in the problem. Some teams tell what they know but don't consider the real question—papers missing several elements are eliminated quickly.
- List all assumptions. The problems are deliberately open-ended, and well-posing them is actually part of the problem. Formulating your assumptions is where you pose the problem—making it simple enough to yield to mathematics yet realistic enough to give a credible result.
- State your conclusions and results clearly and make a precise recommendation.

- Don't just copy the original problem statement, but provide us with your interpretation.

Readability Sometimes the quality of writing is so poor that a judge can't follow or make any sense out of the report.

- Make it clear where in the paper the answers are.
- Many judges find a table of contents helpful.
- Your paper needs to be well organized—can a triage judge understand the significance of your paper in 6 to 10 minutes?
- Keep in mind that your audience consists of modeling experts from academia and industry who have only a short time to get the gist of what you did.

More is not necessarily better. If your paper is excessively long (we have had papers over 100 pp long, not including computer program listing), you should probably reconsider the relevance of all factors that you are discussing. Depending on the round of judging, judges have between 5 and 30 min to read a paper. Do not include a single figure, table, or graph that is extraneous to your model or analysis; such additions just distract the judge from discerning what in your paper is important.

Computer Programs

- Clearly define and explain all variables and parameters.
- For a simulation, a single run isn't enough! You must run enough times to have statistically significant output.
- Always include pseudocode and/or a clear verbal description.

Reality Check

- Why do you think your model is good? Against what baseline can you compare/validate it?
- How sensitive is your model to slight changes in the parameters you have chosen? Teams should undertake sensitivity analysis precisely to build credibility in the model,
- Complete the analysis circle: Are your recommendations practical in the problem context?
- Verify as much as you can. Make sanity checks: Is your answer larger than the number of atoms in the known universe? If it is, should it be?
- Use real data if possible.

Triage Judge Pet Peeves

- Tables with columns headed by Greek letters or acronyms that cannot be immediately understood.

- Definitions and notation buried in the middle of paragraphs of text. A bulleted format is easier for the judge.
- Equations without variables defined.
- Elaborate derivations of formulas taken directly from some source. It is better to cite the source and perhaps briefly explain how the formula is derived. It is most important to demonstrate that you know how to use the formulas properly.

Non-technical report of results If a CEO memorandum, press release, or newspaper article is required:

- Be succinct.
- Include "bottom line and managerial results" answers.
- Do not include methods used or equations.

Resources

- All work needs to be original, or else the sources must be cited (including specific page numbers in documented references). A mere list of books or URLs at the end is not sufficient!
- Teams are allowed to use only inanimate resources—no real people or people consulted over the Internet.
- Surf the Web but cite sites for information that you use.
- Use high-quality references. Peer-reviewed journals, books, and government Websites are preferable to individuals' Websites or blogs.

How to Proceed

- **Read the problem statement carefully.** Words implying actions (design, analyze, compare, etc.) are keys to sections that your paper should contain. Organize the paper into sections corresponding to the parts of the problem; if certain broad topics are required, begin with an outline based on them.
- **Make your paper easy to read.** Number the pages, tables, figures, and equations; check the spelling; and use a large-enough font size.
- **Define terms** that a reader might find ambiguous, particularly any term used in the model that also has a common prose meaning.
- Address sensitivity to assumptions as well as the strengths and weaknesses of the model. These topics should be covered separately in sections of their own. **Go back to your list of assumptions and make sure that each one is addressed.** This is your own built-in checklist aiding completeness; use it.

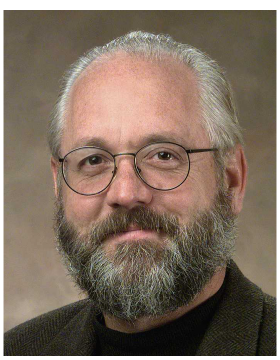
- **Your summary should state the results that you obtained, not just what you did.** Keeping the reader in suspense ("we will develop another model later . . . ") is a good technique in a novel, but it simply frustrates the judges.
- **Do more than is asked.**
- **Write informally, write well.** In many student-written papers, as a colleague puts it, "nobody ever does anything—things just happened." Too common is a chronological narrative in the stilted no-person passive past tense ("Then it was discovered that the data could be fitted by a fifth-degree polynomial . . . "). Much better is a story of first-person present-tense activity ("We fit a fifth-degree polynomial to the data . . . ").

References

Cline, Kelly. 2005. Kelly Cline's Guide to the MCM. `http://web.carroll.edu/kcline/mcm.html` .

Specific instructions on "how to attack the MCM," including a timetable, by a participant who improved steadily (Successful Participant ⟶ Honorable Mention ⟶ Outstanding) over three years and now coaches teams at Carroll College, Montana.

About the Authors



Pat Driscoll is a Professor of Operations Research in the Dept. of Systems Engineering at USMA. He received his M.S. in both Operations Research and Engineering Economic Systems from Stanford University, and a Ph.D. in Industrial & Systems Engineering from Virginia Tech. He is the program director for math electives at USMA. His research focuses on mathematical programming and optimization. Pat is the INFORMS Head Judge for the MCM and the ICM contests.

Jerry Griggs is a graduate of Pomona College and MIT (Ph.D., 1977). Since 1981, he has been at the University of South Carolina, where he is Professor of Mathematics. His research area is combinatorics and graph theory. He has published more than 75 papers and supervised 12 doctoral and 15 master's students. He is the editor-in-chief of the *SIAM Journal on Discrete Mathematics* and serves on the board that oversees the Canada / USA Mathcamp. He has been an MCM judge since 1988 and is the author of four MCM problems.

Mark Parker is Associate Professor of Mathematics at Carroll College. He earned his Ph.D. in Applied Mathematics (combinatorial optimization) from the University of Colorado–Denver. In addition to teaching at Carroll, Eastern Oregon University, and the U.S. Air Force Academy, he spent eight years as a systems simulation and analysis engineer at Lockheed-Martin. After four years as a final-round MCM judge, he enjoys the role of mentor and coach for the Carroll MCM and ICM teams.

Paul Boisen is an Applied Research Mathematician at the National Security Agency. He works a wide range of problems in a dynamic environment and encourages MCM participants to contact him if they join NSA. He was a Fulbright Professor to the Hungarian Academy of Sciences, a Postdoctoral Fellow at MSRI, a graduate student at the University of Chicago (Ph.D. mathematics, 1990), and a student in the first Budapest Semester in Mathematics (Spring 1985). He was a final-round MCM judge 1998–2001 and a head triage judge 2001–2002.

William P. Fox is Professor and Chair of the Dept. of Mathematics at Francis Marion University. He received his M.S. (operations research) from the Naval Postgraduate School and his Ph.D. (operations research and industrial engineering) from Clemson University. He was formerly at the U.S. Military Academy. He makes conference presentations and has co-authored several texts on mathematical modeling. He directs the High School Mathematical Contest in Modeling and has authored numerous MCM / ICM problems.

Mike Tortorella is a Research Professor of Industrial and Systems Engineering at Rutgers University and Managing Director of Assured Networks, LLC. He retired after 26 years from Bell Laboratories as a Distinguished Member of the Technical Staff. He has a Ph.D. (Purdue University) in mathematics. His research interests include stochastic flow networks, information quality and service reliability, and numerical methods in applied probability. Mike has been an MCM judge since 1993 and has particularly enjoyed MCM problems with a practical flavor of mathematics and society. He enjoys amateur radio, the piano, and cycling; he is a founding member of the Zaftig Guys in Spandex road cycling team.

Since the start of the MCM, Paul Campbell has enjoyed editing for *The UMAP Journal* the Outstanding papers and the commentaries on them.

1995: The Helix Intersections Problem

The problem consists of assisting a small biotechnological company in designing, proving, programming, and testing a mathematical algorithm to locate ”in real time” all the intersections of a helix and a plane in general positions in space (see **Figure 1**).

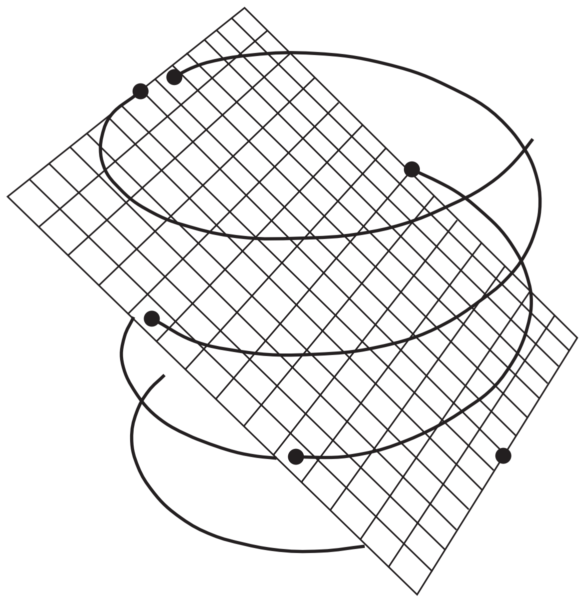


Figure 1. Some intersections of a helix with a plane.

Similar programs for Computer Aided Geometric Design (CAGD) enable engineers to view a plane section of the object that they design, for example, an aircraft jet engine, an automobile suspension, or a medical device. Moreover, engineers may also display on the plane section such quantities as air flow, stress, or temperature, coded by colors or level curves. Furthermore, engineers may rapidly sweep such plane sections through the entire object to gain a three-dimensional visualization of the object and its reactions to motion, forces, or heat. To achieve such results, the computer programs must locate all the intersections of the viewed plane and every part of the designed object with sufficient speed and accuracy. General “equation solvers” may in principle compute such intersections; but for specific problems, special methods may prove faster and more accurate than general methods. In particular, general software for Computer Aided Geometric Design may prove too slow to complete computations in real time, or too large to fit in the finished medical devices being developed by the company. The considerations just explained have led the company to the following problem.

Problem: Design, justify, program, and test a method to compute all the intersections of a plane and a helix in general positions (at any locations and with any orientations) in space.

A segment of the helix may represent, for example, a helicoidal suspension spring or a piece of tubing in a chemical or medical apparatus.

The need for some theoretical justification of the proposed algorithm arises from the necessity of verifying the solution from several points of view. This can be done through mathematical proofs of parts of the algorithm, and through tests of the final program with known examples. Such documentation and tests will be required by government agencies for medical use.

Comments

The Outstanding papers were by teams from Harvey Mudd College, Iowa State University, and Macalester College. Their papers, together with commentaries, were published in *The UMAP Journal* 16 (3) (1995): 209–258.

Problem Origin

The problem was contributed by Yves Nievergelt (Eastern Washington University), from consulting for a small medical technology company in the Western U.S. [1996]. The problem arose in the design of a helicoidal part of a device that doctors and technicians together manufacture to fit the particular measurements of each patient. With x-ray data from the patient loaded into a computer and with a program to compute the requested intersections, doctors and technicians can quickly vary the parameters of the helix, view the helicoidal part superimposed on a model of the patient, and examine critical locations by sweeping a plane section through them.

The mathematically accurate yet medically vague description given in the problem statement typifies a common situation: The small start-up company did not want anyone to know its identity or the object of its research. Such secrecy explains, in part, the dearth of real applications of mathematics in textbooks.

Practitioner's Comments

Pierre J. Malraison (then at Autodesk, Inc.) noted that the Macalester College team used the usual approach to surface-curve intersections: Implicitize the surface (the plane) and parametrize the curve (the helix). The intersections can be found by solving numerically the equation generated when the parametrized form satisfies the implicit equation. The main differences in the Outstanding teams' solutions are in root-finding strategies.

An alternative to numerical solution is to use a rational quadratic parametrization (instead of trigonometric) for the helix and end up with a polynomial function to solve.

The team from Harvey Mudd College used a different approach: Intersect the plane with the cylinder on which the the helix lies, get an ellipse, and then intersect the helix with the ellipse.

Judge's Comments

Contest judge Daniel Zwillinger (Zwillinger & Associates) noted that a plane and a helix can have no intersections, any finite number, or an infinite number (for an infinite helix or a degenerate helix with zero pitch). A program to find all intersection points must respond appropriately in each case.

It is straightforward to write down the parametric equations for the helix, depending on one variable, and the equation of the plane. Substituting the parametric equations into the equation for the plane gives an equation in a single variable. Finding its roots is far simpler than finding the roots of a multiple-variable equation (as some teams proposed).

Bisection is guaranteed to find a zero if endpoints are given—but it is slow. The much-faster Newton's method is the method of choice for nonlinear problems. However, the judges preferred a bisection technique, with provable bounds on the results, to a Newton iteration with no mention of possible convergence problems (for example, near multiple roots). The team from Macalester College used both a Newton iteration and a bisection method (when the Newton method failed).

The papers addressed computational speed in various ways: computer time per intersection point, computer time saved compared to a more general mathematical solver (such as Mathematica), or computational complexity of the algorithm.

Several teams restricted the problem to a finite helix, including those from Harvey Mudd College and Iowa State University. Using a finite-area sweeping plane was considered by the team from Harvey Mudd College. Additionally, one team considered more general helices, such as a spiral drawn on a cone.

References

Nievergelt, Yves. 1996. Intersections of planes and helices, or lines and sinusoids. *SIAM Review* 38 (1)(March 1996): 136–145.

Planes and Helices

Samar Lotia
Eric Musser
Simeon Simeonov
Macalester College
St. Paul, MN 55105
{slotia, ssimeonov}@macalstr.edu
emusser@prodea.com

Advisor: A. Wayne Roberts

Introduction

We are asked to design, implement, and test a mathematical algorithm that locates in real time all of the intersections of a helix and a plane located in general positions in space. In addition, we must prove that our algorithm is mathematically and computationally correct.

Assumptions

- **"Real time."** We assume that "real time" means that the time to solve a reasonably "difficult" problem must be very small. For example, an algorithm that is used for the back end of a Computer Aided Geometric Design program must not impose unacceptable delays on the user. When the user repositions the helix or plane, the user expects an immediate refresh of the screen—usually in only a fraction of a second. For other applications, our algorithm should take little time compared to the calling program. To meet these requirements, the algorithm must have a time complexity linear in the number of intersections; our algorithm is indeed linear.

- **Helix/Plane.** We assume the strict mathematical definition of a single helix: that it is infinite and nonelliptical, and that it "wraps around" a cylinder. We believe that our algorithm will work for other "helices," such as those having elliptical bases; but we have not tested these cases to any extent. We also assume that the plane extends infinitely.

- **Correctness.** We assume that twelve digits of precision is acceptable for all calculations. This is sufficient for most biotechnological applications. Twelve digits means that we have error less than the radius of some atoms [Chang 1994]. For some applications, engineers often expect fewer digits of accuracy; so our program allows the user to change the desired precision.

Summary of Approach

Our approach to the problem involves

- definition of design requirements,
- development of a mathematical model for the problem,
- design and implementation of an algorithm,
- debugging and testing, and
- evaluation.

Definition of Design Requirements

The algorithm will be used as a support routine for mission critical processes, where its failure to produce correct results, or failure to produce results on time, could have dire consequences. This fact leads us to the following design requirements, in decreasing order of importance:

- **Correctness.** The algorithm must produce correct results.
- **Robustness.** The algorithm must handle exceptions well and not terminate abnormally.
- **Performance.** The algorithm must execute in real time.
- **Efficiency.** The algorithm must spare system resources, as long as the above three requirements are not adversely affected.
- **Flexibility.** The algorithm must allow users to formulate the problem in different ways, e.g., it must allow more than one way of defining a plane or a helix. Also, users must be able to fine-tune the algorithm to improve performance.
- **Portability.** The algorithm must be machine-independent, written in a common programming language, and easy to transfer to a different programming language of choice (e.g., to include it in an embedded system).

Because algorithms execute on physical entities (computers) that have some finite working precision, it is possible that a computationally correct algorithm may produce incorrect results due to roundoff and compound errors. In our case, two types of errors are possible: skipping an existing root and reporting a root that does not exist. It is difficult to claim that one type of error should be preferred to another; we choose, if possible, to minimize the second kind of error given the possibility of some error of the first type.

Development of a Mathematical Model

Initial Development

The General Case

Having examined the problem in general Cartesian coordinates, we found that a graphical or vector analysis–related approach would fail us, in the sense that it would be incredibly hard to program. Hence, we attempted to reduce the problem to an algebraic problem, since algebraic techniques are generally much more suited to programming. In effect, we "projected" the helix onto the plane in the following manner.

Consider the general parametric equations of a helix in space (for derivation, see the **Appendix**):

$$\begin{aligned} x &= a_{11}\cos\alpha t + a_{12}\sin\alpha t + a_{13}t + a_{14} \\ y &= a_{21}\cos\alpha t + a_{22}\sin\alpha t + a_{23}t + a_{24} \\ z &= a_{31}\cos\alpha t + a_{32}\sin\alpha t + a_{33}t + a_{34}. \end{aligned}$$

The equation of a plane in space may be written as

$$ax + by + cz - d = 0.$$

Our transformation replaces x, y, and z on the left-hand side in this equation with the corresponding parametric forms of the helix, thus returning an expression in t, which we call $f(t)$:

$$f(t) = A\cos t + B\sin t + Ct - D,$$

where A, B, and C are appropriately transformed coefficients. The α in the $\cos$ and $\sin$ terms has been incorporated into C via a change of parameter from t to t/α.

The task now is to solve the equation $f(t) = 0$. After perusing relevant literature, we concluded that this equation must be solved numerically [Plybon 1992]. Hence, we developed a numerical technique that, given the parameters A, B, C, and D, attempts to locate all the roots of the equation. Many well-documented algorithms guarantee convergence to roots—given certain bounds on the problem—and give an easily computable bound on the error in the result. The numerical technique that we employ is heavily influenced by information that we have about the equation $f(t) = 0$. For instance, we know that the extrema of the function (if any) occur periodically, and we use this fact at several stages of our method, hence ensuring an efficient algorithm. We essentially built a robust, *strong* equation solver (one that uses problem-specific information to maximize the efficiency of the algorithm).

The first step in our approach is to examine the general form of the function, as in **Figure 1**. We note that all roots must be located between two extrema that are on opposite sides of the t-axis (assuming a continuous function). The only

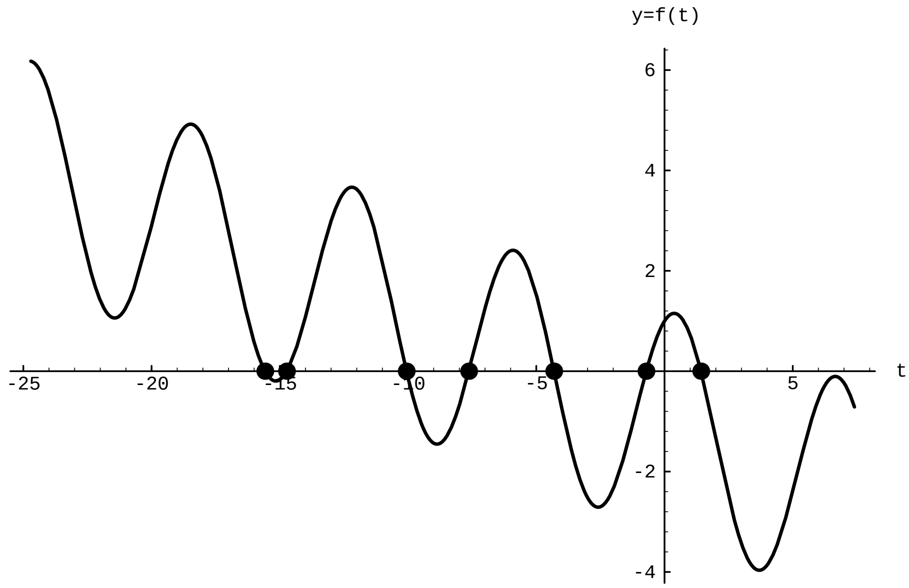


Figure 1. General functional form.

other case, which we handle separately, is when a root and an extremum or inflection point occur simultaneously.

We begin by locating the minima and maxima of the function $f(t)$, which are periodic with period 2π. These are found by solving the equation

$$f'(t) = -A\sin t + B\cos t + C = 0.$$

From

$$\cos t = \frac{A\sin t - C}{B},$$

by using $\cos^2 t + \sin^2 t = 1$ we find

$$\sin t = \frac{ac \pm b\sqrt{a^2+b^2-c^2}}{a^2+b^2}.$$

However, this method returns some extraneous roots because of the squaring (just as does squaring $t = 1$ and solving $t^2 = 1$). These are discarded via a simple test, namely, substituting the values back into $f'(t)$ and checking whether or not the derivative is indeed zero.

We then interpolate the root by connecting the two extrema via a line segment (as in **Figure 2**) and pass the interpolated value to our root-finding algorithm.

We must now judiciously choose a value of t that guarantees roots in its immediate neighborhood. We choose the value $t_0 = D/C$, as we are ensured that—if roots exist—there is one within 2π of this value of t (see the **Appendix** for a detailed argument).

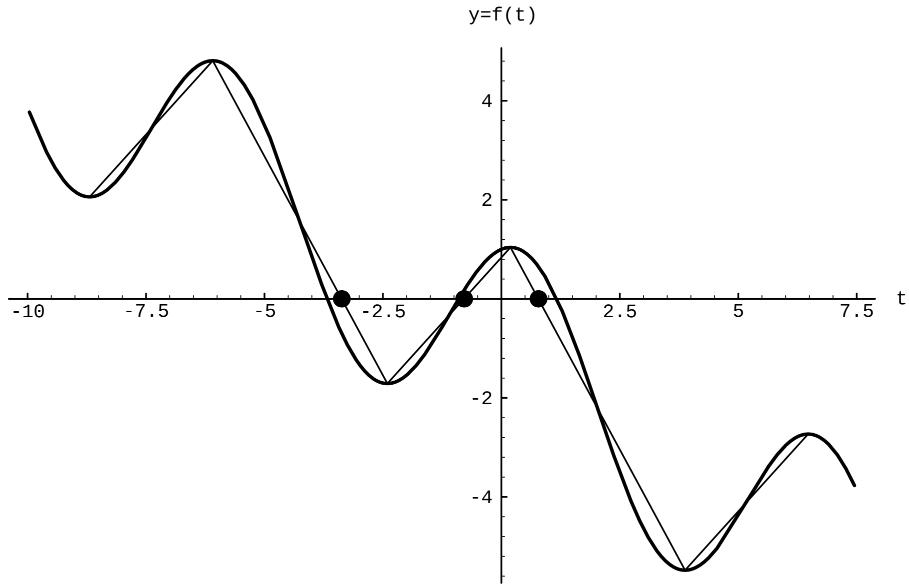


Figure 2. The interpolation method.

Certain Special Cases

If the coefficient C is 0, then the function $f(t)$ is periodic and oscillates to within $\sqrt{A^2+B^2}$ (the maximum possible value of $A\cos t + B\sin t$) about the line $g(t) = D$. Thus, if $|D| > \sqrt{A^2+B^2}$, then the function never intersects the t-axis and we have no roots; the plane is parallel to the helix and outside the "reach" of its radius. In such a case, our program returns the message "No Roots." If, on the other hand, $|D| \leq \sqrt{A^2+B^2}$, then we have infinitely many roots; this case is handled appropriately.

Another important case is that of only one root (see **Figure 3**). This occurs when $a^2 + b^2 - c^2 \leq 0$. Then the equation has either one real solution or none. If a solution is found, it is at an inflection point. This condition is recognized by our algorithm and appropriately dealt with by calling the bisection method instead of the usual Newton-Raphson (which exhibits very slow convergence when dealing with simultaneous occurrence of roots and extrema/inflection points). The bisection method guarantees convergence as long as we bracket the root properly. We are confident that we do so, as we give bisection an interval of radius 2π about the point $t_0 = D/C$. The bisection method achieves our prescribed goal of 12-digit accuracy in no more than 44 iterations.

Algorithm Description

To facilitate understanding, we provide four levels of abstraction in the description of our algorithm. At the top level, we use a linearized flowchart that shows the main subproblems that need to be solved. [EDITOR'S NOTE: Because of space considerations, we do not reproduce the flowchart.] Parallel

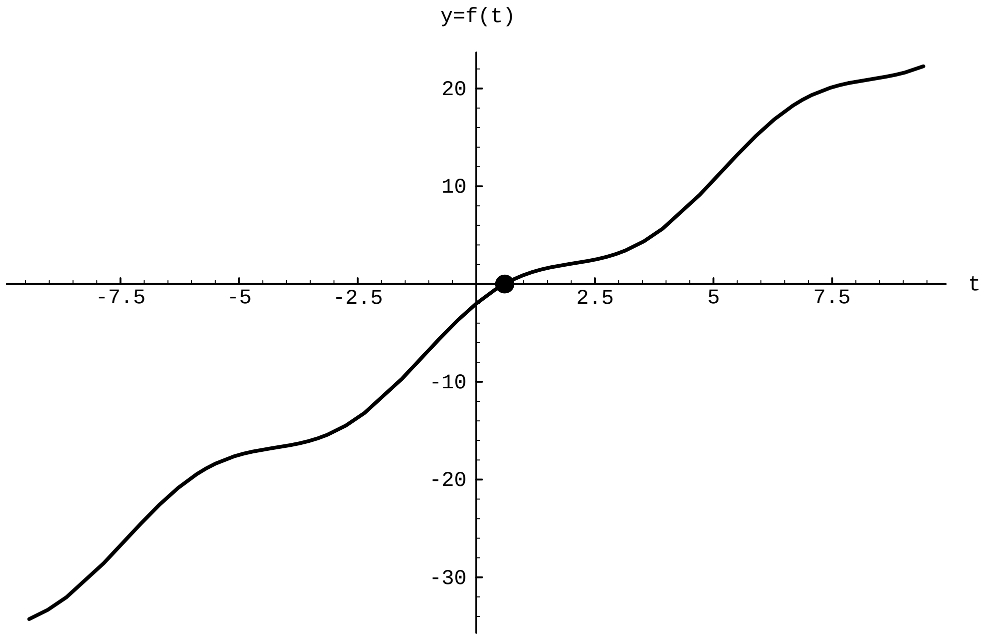


Figure 3. The single-root case.

to the flowchart, at the second level of abstraction, we offer comments that provide more detail of the workings of the algorithm. They refer to the third level of abstraction, mathematical proofs and detailed explanations. Finally, at the lowest level of abstraction is the C++ code of our program, which includes many comments with Mathematica code and references to relevant literature.

Before we go into the details of the algorithm implementation, we mention some general conventions.

- **Input**
  - Planes can be defined by the user in three ways: by general Cartesian equation of the form $ax + by + cz = d$, by two vectors and a point, or by three points.
  - Helices can be defined by the user in two ways: by general parametric equations, or by the three Eulerian angles and the translation vector that map the z-axis to the central axis of the helix.
- **Output**
  - If the helix does not intersect the plane, no roots are returned.
  - If there are infinitely many solutions (a case in which the plane is parallel to the helix axis), sufficient information is provided so that the user can generate all of the intersection points.
  - Otherwise, a structure containing the x, y, and z coordinates of the points of intersection is produced.

- **Accuracy of Estimation.**

  The default working precision of calculation in our C++ program is twelve digits, but it can be changed by modifying a single variable in the code. The maximum working precision is limited by the floating point precision of the computer.

- **Portability.**
  - Our algorithm is implemented in ANSI C++, ensuring portability across most computing platforms. The code does not use machine-dependent features, and it could be translated to any procedural language.

Testing and Quality Control

We devoted more than half of our algorithm design and implementation efforts to testing. We checked the correctness of ideas and implementations at four different levels:

- **Math Model.** All transformations and function forms that we used were generated symbolically using Mathematica's standard features and the `Vector Analysis` and `Rotations` packages. All symbolic solutions to equations were checked using Mathematica. Whenever possible, we simplified expressions, sometimes by applying trigonometric substitution rules manually.

- **Algorithm Design.** Our root-finding procedure was carefully chosen always to find a bracketed root (if necessary, by invoking bisection).

- **Implementation.** We applied the function `CForm` to convert Mathematica expressions to C code when transferring expressions to our implementation. That minimized the chances of erroneous expression entry. The root-finding procedure that we use was taken from Plybon [1992]. The procedure was independently tested against the built-in Mathematica routine `FindRoot`, which implements a combination of Newton-Raphson and the secant methods [Wolfram 1991]. Our procedure never failed to find a root and never reported a root where there was none. In several cases where `FindRoot` failed, our procedure correctly managed to find a root.

- **Runtime.** We performed three different types of checks on the output of our program:
  - We generated more than 50 functions of the form $f(t) = A\cos t + B\sin t + Ct - D$ and checked whether our program correctly found their roots. We implemented a set of Mathematica routines that finds the roots of the equation $f(t) = 0$ by the same algorithm as our program but with higher accuracy. In all test cases, the output of our program agreed with Mathematica's output, suggesting that roundoff error is not a major problem in our implementation. In most cases, we visually inspected the graph

of $f(t)$ to ensure than no roots were missed and that no false roots were introduced. The tests included a mix of test cases including none, one, many, and infinitely many roots. We considered potentially problematic cases, such as roots at tangency points and roots at inflection points. We explored and tested all control paths of the algorithm. Debugging output was generated and investigated carefully.

- We inputted more than fifty helices and planes in various input formats and used our algorithm to find the coordinates of the intersection points between them. Then for each test run we used Mathematica to do a 3-D plot of the plane, the helix, and the intersection points (see **Figure 4**). We inspected the 3-D plots from various viewpoints to ensure that no intersection points were missed and that no extraneous points were plotted. Our program passed all tests.
- In the testing, often we were uncertain about the actual location of the intersection points. So we designed an additional battery of tests to check the results of our program against a known (sub)set of the intersection points. We obtained the known intersection points by starting with an arbitrary helix and defining an intersecting plane either by choosing three points on the helix or by choosing a point on the helix and an arbitrary vector. In the first case, we knew the coordinates of at least three intersection points. In the second case, we could position the plane so as to experiment with different patterns of intersection. We then checked the results of our program against subset of known roots. In all of the more than 50 test cases, our algorithm performed correctly.

In *all* test cases, our program performed successfully. This result—added to the fact that our algorithm is closely based on a rigorously proven mathematical model—gives us a high degree of confidence that our program is indeed computationally correct. Also, we have seen no evidence that roundoff or compound errors are a significant source of error.

Evaluation

Correctness

Our tests have shown that mathematically and computationally our algorithm is correct. Numerically, however, problems can arise. The possibility of compounding error and loss of precision is inevitable in some cases (though rare).

Compounding error is introduced in routines that map our original helix onto a helix located about the z-axis (i.e., computing the coefficients A, B, C, and D, for $f(t)$). We are uncertain of the error bounds on these calculations, but comparisons with high-precision Mathematica runs show that the error is small (less than 10^{-12}).

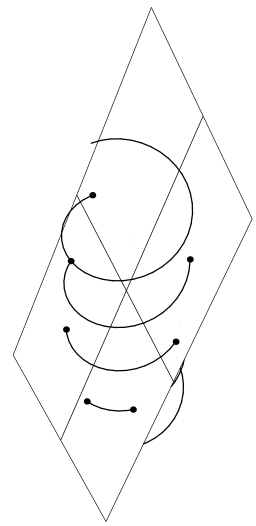


Figure 4. Our solutions plotted on the given helix and plane.

Fortunately, given correct values of A, B, C, and D, we can guarantee that the roots found are accurate to the desired working precision. This is because we use Newton-Raphson and the bisection method as our root-finding techniques. With Newton's method, it is possible to get some idea of absolute error of a root x_0 simply by looking at the value of $f(x_0)$ (it should be 0 if we have a root). Also, error is not compounded with Newton's method; each iteration actually decreases the error. Newton-Raphson converges quadratically [Plybon 1992], and near a root the number of significant digits approximately doubles with each step.

Detrimental error can also enter the problem when calculating the brackets for each root. If error in numerical computation places a bracket on the wrong side of a root, that root may be lost. We have never encountered such a case but expect that cooked-up data could produce such an error. A possible remedy (not guaranteed to work in all cases) is first to approximate the bracket using our exact formulas and then to run a root-finding method on them to reduce the error. We did not implement this strategy, because we feel that the chance of this error occurring does not warrant the loss in speed that will occur.

Further, in searching for solutions of $f(t) = 0$, if two roots are extremely close, they can unfortunately be mistaken as one. However, if two extremely close roots are found, it may not even make sense to consider them as distinct, since they may be the product of numerical error.

Robustness

Our algorithm checks extensively for exception cases and will not terminate abnormally due to a computation error or lack of system resources. We implemented checks for special cases that in effect trap errors. One example: We check for an infinite number of roots before we search for roots. We handle "special" cases such as tangencies, inflection points, double roots, etc.

Our algorithm ensures that all roots are bracketed. This prevents Newton's method from accidentally going off and finding another root. Our implementation of Newton's method incorporates the bisection method in cases where Newton's does not converge fast enough or Newton's method departs the bracketed interval. The bisection method is guaranteed to find a root within a bracketed interval [Plybon 1992].

In our algorithm, there is an inherent limit on the number of possible roots; but this limit can be easily increased or even eliminated (allowing memory of the computer to be the only limitation).

Performance/Efficiency

The performance of our algorithm is linear in the number of intersections. This is because we do a single root-find for each intersection (including bracketing). The complexity of a root-finding method is dependent on the how the function is shaped and the number of digits desired. Typically, when Newton's method is used, 5 or 6 iterations are required to find each root to 12 digits of precision, while the bisection method can be shown to take fewer than 44 iterations.

For mapping the helix, finding first-order roots, etc., the time required is relatively constant. Therefore, the time complexity of the algorithm is dominated by the number of intersections.

In addition, our code is efficient in use of space, since the space complexity is also linear in the number of intersections.

Suggestions for Improvement

As the general organization of our algorithm is closely based on a mathematical model, we believe that it is not possible to improve it without a thorough revision of the underlying methodology. However, the *implementation* of the algorithm can be improved in several ways:

- One can attempt to modify the evaluation of expressions so as to reduce the compound and roundoff errors. This often necessitates understanding and use of architecture-specific features of the processor on which the algorithm is executing, thus limiting portability.

- One can attempt to find bounds for the error introduced during the calculation of the A, B, C, and D coefficients for $f(t)$. An estimation of the relationship between this error and the error generated by the root-finding algorithm will also be helpful.

- A useful yet hard-to-implement feature would be the inclusion of internal validation routines that improve correctness and robustness by monitoring for unacceptable computational errors while the algorithm is executing. Such routines would adversely affect performance.

- The input procedures could potentially be extended to include alternative definitions of a helix. However, our explorations of the relevant literature yielded no definitional forms different from the ones that we use.
- The Mathematica routines used in the testing process could be extended to infinite-precision calculation, running in batch mode to check at random the correctness of solutions provided by the real-time algorithm.
- The algorithm could be extended to handle other types of helices: alpha helices, double helices, etc. The underlying methodology of the algorithm should remain unchanged.

Appendix

General Parametric Equations for a Helix

Consider a general 3×4 rotation-translation matrix:

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{33} & a_{34} & a_{34} \end{bmatrix}$$

and the "unit" helix (in vector form)

$$(\cos(\alpha t - t_0), \sin(\alpha t - t_0), t).$$

Applying the matrix produces any general helix:

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{33} & a_{34} & a_{34}. \end{bmatrix} \begin{bmatrix} \cos(\alpha t - t_0) \\ \sin(\alpha t - t_0) \\ t \\ 1 \end{bmatrix},$$

giving the general parametric equations of a helix in space:

$$\begin{aligned} x &= a_{11}\cos(\alpha t - t_0) + a_{12}\sin(\alpha t - t_0) + a_{13}t + a_{14} \\ y &= a_{21}\cos(\alpha t - t_0) + a_{22}\sin(\alpha t - t_0) + a_{23}t + a_{24} \\ z &= a_{31}\cos(\alpha t - t_0) + a_{32}\sin(\alpha t - t_0) + a_{33}t + a_{34}. \end{aligned}$$

Upon expanding the $\sin$ and $\cos$ terms, we obtain the same format as presented in the text of the paper.

Why $t_0 = D/C$?

The equation

$$f(t) = a\cos t + B\sin t + Ct - D$$

can be thought of as the intersection of a helix with parametric equations

$$x = \cos t, \qquad y = \sin t, \qquad z = t$$

with a plane with equation

$$Ax + By + Cz = D.$$

As far as the root-finder is concerned, we have a "vertical" helix with radius 1 and the plane as above. Thus, we say that the point at which the helix's central axis (also the z-axis for the root-finder) meets the plane $Ax + By + Cz = D$ is the point around which the roots are distributed almost symmetrically.

The justification for this claim is as follows. The intersection of a plane and a cylinder in space is an ellipse; as the helix lies on a cylinder, its intersections with a plane must lie on an ellipse. The center of the ellipse is the point of intersection of the plane and the helix's central axis. In **Figure 5**, we show the cylinder that the helix sits on, the ellipse of intersection with the plane, and the helix itself. The curve represents the ellipse of intersection and the lines are the helix. The whole picture shows the cylinder "unfolded."

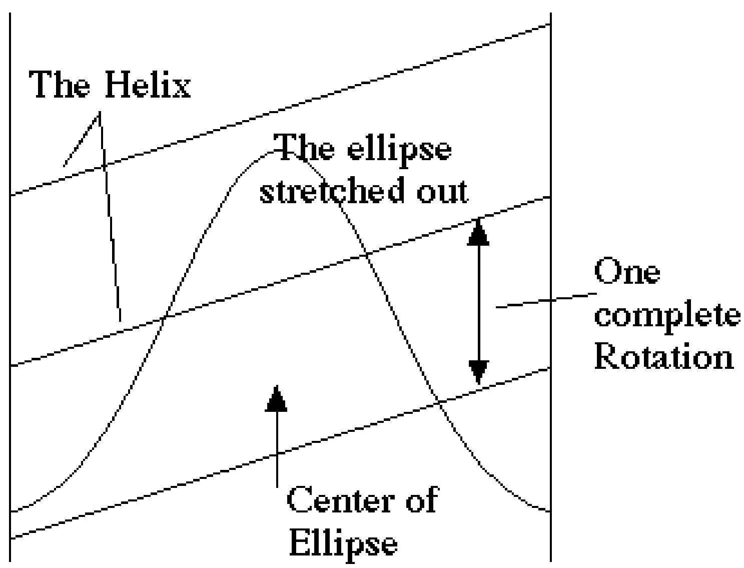


Figure 5. Representation of the intersection of the helix and the plane.

It can be shown that if any roots exist, they must do so within one complete rotation of the center of the ellipse. Hence, we use the center as the starting point for our root-finder.

References

Canale, Raymond P., and Steven C. Chapra. 1985. *Numerical Methods for Engineers.* Reading, MA: Addison-Wesley.

Chang, Raymond. 1994. *Chemistry.* 5th ed. New York: McGraw-Hill.

Englefield, M.J. 1987. *Mathematical Methods for Engineering and Science Students.* Edward Arnold.

Press, William F., et al. 1990. *Numerical Recipes in Pascal: The Art of Scientific Computing.* New York: Cambridge University Press.

Plybon, Richard F. 1992. *Applied Numerical Analysis.* Boston: PWS-Kent.

Wolfram, Stephen. 1991. *Mathematica: A System for Doing Mathematics by Computer.* 2nd ed. Reading, MA: Addison-Wesley.

1996: The Contest Judging Problem

When determining the winner of a competition like the Mathematical Contest in Modeling, there is generally a large number of papers to judge. Let's say that there are $P = 100$ papers. A group of J judges is collected to accomplish the judging. Funding for the contest constrains both the number of judges that can be obtained and the amount of time that they can judge. For example, if $P = 100$, then $J = 8$ is typical.

Ideally, each judge would read all papers and rank-order them, but there are too many papers for this. Instead, there are a number of screening rounds in which each judge reads some number of papers and gives them scores. Then some selection scheme is used to reduce the number of papers under consideration: If the papers are rank-ordered, then the bottom 30% that each judge rank-orders could be rejected. Alternatively, if the judges do not rank-order the papers, but instead give them numerical scores (say, from 1 to 100), then all papers falling below some cutoff level could be rejected.

The new pool of papers is then passed back to the judges, and the process is repeated. A concern is that the total number of papers that each judge reads must be substantially less than P. The process is stopped when there are only W papers left. These are the winners. Typically, for $P = 100$, we have $W = 3$.

Your task is to determine a selection scheme, using a combination of rank-ordering, numerical scoring, and other methods, by which the final W papers will include only papers from among the "best" $2W$ papers. (By "best" we assume that there is an absolute rank-ordering to which all judges would agree.) For example, the top three papers found by your method will consist entirely of papers from among the "best" six papers. Among all such methods, the one that requires each judge to read the least number of papers is desired.

Note the possibility of systematic bias in a numerical scoring scheme. For example, for a specific collection of papers, one judge could average 70 points, while another could average 80 points. How would you scale your scheme to accommodate for changes in the contest parameters (P, J, and W)?

Comments

The Outstanding papers were by teams from Fudan University (Shanghai, China), Gettysburg College, St. Bonaventure University, University of Science and Technology of China (Hefei, China), and Washington University. Their papers, together with commentaries, were published in *The UMAP Journal* 17 (3) (1996): 283–348.

Problem Origin

The problem was contributed by Daniel Zwillinger (Zwillinger & Associates).

Practitioner's Comments

Veena Mendiratta (Lucent Technologies) noted that what made the problem interesting and challenging were the less well-defined aspects of the problem: How was bias of judges handled? How was it ensured that the best $2W$ papers were not screened out early? For a "best" selection scheme, teams used methods such as rank-ordering, numerical scoring, and bias estimation.

Almost all papers developed a basic model for the ideal case of no judge bias: Every judge rank-orders or scores the papers in accord with an overall absolute rank-ordering. This model sets a lower bound for the total number of reads. The stronger papers went significantly beyond the ideal case to include:

- Explicit modeling of judges' bias, addressing the issues of systematic bias and the variance of accidental errors in scoring.
- Estimating statistical bounds on the probability of failure to pick the best W papers out of the top $2W$ papers.
- Realizing the importance of more judges reading the papers remaining in the later screening rounds and including this factor in their models.
- Clear statement of results in terms of total number of paper readings, confidence level of the results, and sensitivity of results to the model parameters.

The Gettysburg College team minimized the probability of eliminating the W best papers in the first round by having two judges read each paper in that round. This team modeled judge error through a functional relationship between the probability of judge error in ranking (with respect to the absolute ranking) and the distance between papers on the absolute scale. The team from the University of Science and Technology of China used Bayesian estimation to address systematic bias in scoring and modeled the error probability as a function of the percentage of papers eliminated in each round. The Fudan University team showed statistically the conditions under which the "ideal" model can work. The Washington University team modeled the judge bias and error and, after each round of judging, used new bias estimates to calculate confidence intervals that determine the number of papers rejected. The St. Bonaventure team implemented a distribution scheme to ensure that judges do not receive the same paper more than once and that one judge does not receive the top $2W$ papers.

Judge's Comments

Contest judge Don Miller (St. Mary's College, IN) cited other common applications for the techniques developed by the teams: awarding scholarships and screening applicants for a position.

Assuming absolute rank-ordering made the problem seem deceptively easy. With 100 papers and 8 judges, there are methods to find the top 3 papers (in order) with each judge reading at most 14 papers and with at most 109 papers read. Some papers simply developed a heuristic solution; others recognized that the assumption was unrealistic but chose to model the problem as stated, since that is what was requested. Some who used this assumption clearly didn't believe it, since they rejected a simple merge-sort in favor of more complicated algorithms that denied existence of the assumption. Still others attempted to refine the problem using theories ranging from graph theory to fuzzy sets. Some of the better bias-elimination refinements included matrix reduction, regression with error terms, and scoring normalization with specified probability distributions.

Common in problems are unrealistic assumptions that would make any model based on them of minimal use. Thus, a modeler must evaluate critically all assumptions and, if necessary, refine the problem to a realistic one. Judges viewed absolute rank-ordering as such an assumption. In the absence of an opportunity to ask for elaboration, the modeler should answer the question as stated and then refine it with realistic assumptions.

The judges felt that the best models solved the basic problem for 100 papers, 8 judges, and 3 winners, then produced a refinement with adequate complexity to model the process accurately but with enough simplicity for the model to be useful. The ideal paper would solve the basic problem and demonstrate that the solution was optimal (or close to optimal). It would then generalize to accommodate different numbers of papers, judges, and winners. It would address judge bias and measure the success rate of the algorithm as a function of some quantitative measure of judge bias. It might then examine alternative algorithms, finding a relation between levels of judge bias and the success rate of the algorithms. It would assess the strengths and weaknesses of these methods, while addressing all the points requested in the problem statement.

The Outstanding papers were distinguished by how they addressed judge bias. The team from the University of Science and Technology of China used Bayesian statistics, with a normal prior, to adjust for judge bias, then ran simulations to test for sensitivity. Normal distributions, a common assumption, were also used by the Washington University team for both the intrinsic score of the paper and judge bias. A distinguishing feature of the paper from Fudan University was its stability analysis for different levels of judge bias.

Contest Director's Comments

Contest Director Frank Giordano (COMAP) discusses current practice in handling MCM submissions on pp. 11–14.

Select the Winners Fast

Haitao Wang
Chunfeng Huang
Hongling Rao
Center for Astrophysics
University of Science and Technology of China
Hefei, China

Advisor: Qingjuan Yu

Summary

Assuming that judges are ideal, we provide a model to determine the top W papers in almost the shortest time. We use a matrix record the orderings that we get from judges, and we reject as many papers as possible after each round.

We then consider real-life judges and estimate the probability that the final W papers contain a paper not among the best $2W$ papers.

Furthermore, considering the possibility of systematic bias in a scoring scheme, we improve the model by using a Bayesian estimation method, which makes it possible to some extent to compare different judges' scores.

We performed many computer simulations to test the feasibility of our model. We find that our model would be improved by increasing the number of papers selected from the first round. We also made a stability analysis by altering P, J, W and got an empirical formula to predicts the total time of judging.

We used data from real life to test our model and got a perfect result: For $P = 50$, $J = 3$, and $W = 2$, we get the first- and third-best papers with our scheme; with $W = 3$, we got the top three papers.

We conclude by summarizing a practicable and flexible scheme and offering some suggestions, and estimating the budget with an empirical formula.

Assumptions

- The judges are equal. None is more authoritative than the others.
- When a judge is evaluating a paper, the judging result is not influenced by adventitious factors, such as taking bribes.
- The time that a judge takes is proportional to the number of papers to read.

- There exists an objective criterion with which we can tell which of two papers is "better." Therefore, we can use an absolute rank-order or absolute scores to describe the quality of the papers measured by the criterion.

- The absolute rank-order is transitive: If A is better than B and B is better than C, we can say A is better than C.

Analysis of Problem

Our primary goal is to include the top W papers among the "best" $2W$ papers.

A subsidiary goal is that each judge read the fewest possible number of papers. We interpret this goal into two points:

- It is the duration of the whole judging process, the total time for all rounds, that is constrained by funding. If the time for a round is how long it takes the judge who has the most papers to read, it is wise to distribute the papers to the judges as evenly as possible in each round.

- We want to get as much information as possible.

The two usual methods of judging are rank-ordering and numerical scoring. Systematic bias is possible in a scoring scheme, that is, each judge may have a subjective tendency in scoring, which results in incomparability among scores given by different judges. However, it is reasonable to believe that the scores that the same judge gives to different papers are comparable, even if they are obtained in different rounds. Therefore, compared with a rank-ordering method, scoring is a more meaningful way to record the results for papers judged in earlier rounds. We use a scoring scheme instead of a rank-ordering scheme in our later model, so a paper need not be read more than once by the same judge. Note that we do not compare the scores of different judges directly, that is, we mainly use scores to obtain a rank-ordering.

We first consider the simplified problem with the significant assumption that the ordering from each judge's evaluation coincides with the absolute ordering. In this event, we can definitely find the best W papers. Furthermore, we can optimally adjust the allocation of papers in every round to get an efficient scheme.

But judges in real life cannot rate the papers with perfect precision. For example, a paper with absolute rank 7 (we denote it $P_{(7)}$) may get a higher score than $P_{(6)}$ from a judge. We call that *misjudgment*. Misjudgments prevent us from getting the best W papers, so their effect must be taken into account.

There are also subjective differences among the scorings of different judges. For example, for the same two papers, one judge may give 80 and 83, while another gives 65 and 72. If we know the distribution of each judge's scores, we can to some extent compare scores given by different judges. The real distribution for each judge is unknown, so we have to use estimates.

Table 1.

Notation.

Symbol	Meaning
P	total number of papers
W	number of winners
J	number of judges
T	total judging time (or number of papers that can be judged in the time)
P_i	paper i
$P_{(i)}$	paper with the absolute rank of i
S_i	the absolute score for paper i
$P_i > P_j$	paper i is better than paper j in absolute rank-order
$P_i(A) > P_j(A)$	paper i is better than paper j in judge A's opinion
R_i	number of papers currently known to be better than P_i
ORD	matrix of currently known relations between pairs of papers
$\lceil x \rceil$	the smallest integer not less than x
$N(\mu_0, \sigma_0^2)$	normal distribution with mean μ_0 and standard deviation σ_0
σ_1	standard deviation of the judges' scoring
μ_j, σ_j	mean and standard error of judge j's scoring
$\hat{\mu}_j, \hat{\sigma}_j$	estimated values of μ_j, σ_j
P_{error}	probability of error occurring

Design of the Model

Top W in the Least Time

Ideally, the ordering in each judge's opinion coincides with the absolute ordering, expressed mathematically by

$$P_i(A) > P_j(A) \Leftrightarrow P_i > P_j.$$

So, based on the transitivity of the absolute score, if $P_i(A) > P_j(A)$ and $P_j(B) > Pk_(B)$, we can say that $P_i > P_k$.

To find the top W as soon as possible, as many papers as possible should be rejected after each round. So if there are W papers or more in the current paper pool that are better than P_i , reject P_i.

In the first round, P papers are dispatched to J judges evenly. After performing the above rejection rule, each judge selects W papers. In later rounds, how do we dispatch the remaining $W \cdot J$ papers to judges to obtain the greatest number of new orderings from each round?

Let us consider the simple case in which $W = 2$, with $P_1 > P_2$ and $P_3 > P_4$ known after the first round. If we compare P_2 with P_4 (or P_1 with P_3), no matter what the result is, we can always gain an extra relation (if, say, $P_2 > P_4$, the extra relation is $P_1 > P_4$). If we compare P_1 with P_4 (or P_2 with P_3), on some occasions no extra relations can be obtained. A similar result holds for $W = 3$.

This example indicates a fact: If we use *current rank* R_i to denote the number of papers known to be better than P_i from current known information, we should try to distribute the papers with close current rank to the same judge in order to get more relations in a round.

Use matrix ORD to describe known orders. Define ORD_{ij} by

$$\mathrm{ORD}_{ij} = \begin{cases} 1, & \text{if } P_i > P_j; \\ -1, & \text{if } P_j > P_i; \\ 0, & \text{if } P_j = P_i; \\ \infty, & \text{if } P_i \text{ and } P_j \text{ have not been compared by any judge.} \end{cases}$$

At the beginning of a round, we dispatch papers to judges and judges give each paper a score. We then find every P_i and P_j that have scores from the same judge in the finished rounds and fill ORD_{ij} and ORD_{ji}. We replace ORD with its transitive closure [Wang 1986], which, put simply, adds all the indirect order gained from ORD_{ij} into the matrix ORD. At the end of each round, for each paper P_i, calculate R_i from ORD and reject P_i if $R_i \geq W$. Repeat the above process until the final W papers are left.

Consider Misjudgment

By *misjudgment*, we mean that the final W papers are not the best W. If the final W papers contain a paper not among the best $2W$, an *error* occurs.

Assume that for a paper with an absolute score of μ_1, the score given by a certain judge is a random number following a normal distribution $N(\mu_1, \sigma_1^2)$. The standard deviation σ_1 is the parameter that describes the degree of precision in a measurement. Misjudgment originates from the deviation of a judge's scoring from the absolute score, as **Figure 1** shows. The shaded area in **Figure 1** shows the misjudgment area.

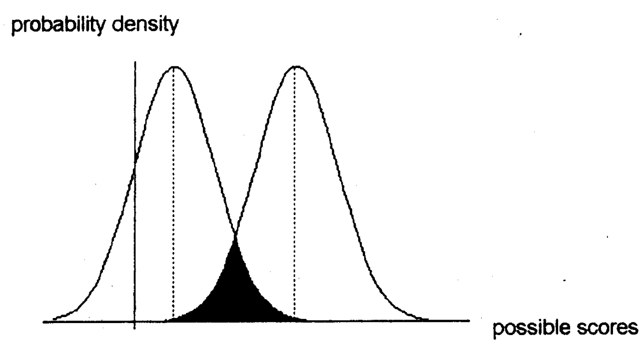


Figure 1. Possible score distributions for two papers.

There must be a distribution of the absolute scores of all the papers. We assume that it is a normal distribution $N(\mu_0, \sigma_0^2)$, so that the ratio σ_1/σ_0 reflects the judge's ability to distinguish the quality of these papers and also determines the probability of misjudgment. Using the basic model, given P, J, W, and σ_1/σ_0, we can estimate the probability of error (P_{error}). If the probability is small enough, we can expect the model to provide the desired result.

Taking the random feature of scoring into account, some conflict is likely to happen, such as $\mathrm{ORD}_{ij} = 1$ but judge A's scores show $P_j(A) > P_i(A)$. One way to solve the conflict is to find all judges who have read both P_i and P_j, sum up the scores given to P_i and P_j by these judges, and determine a new ORD_{ij} by comparing the two sums.

Systematic Bias Among Judges

Considering differences among the scoring tendencies of different judges (systematic biases), it is undesirable that each judge select out the same number of papers in the first round, for then it will be more likely that excellent papers will be rejected in the first round.

Instead, when the first round of judging is over, we input the scores of each group of papers into computer, which gives the estimate of each judge's parameters (mean score and standard deviation) and computes each group of papers' score threshold for rejecting papers corresponding to a certain absolute level. This way, excellent papers have less possibility of being rejected in the first round. Estimating the parameters of all the judges enables us to compare the scores from different judges to some extent.

We use Bayesian estimation [Box and Tiao 1973] to determine the estimate of judge j's parameters (μ_j, σ_j). Suppose that judge j gives scores $S_1, \ldots, S_n$ to papers $P_1, \ldots, P_n$. We use the method of maximum likelihood to estimate σ_j:

$$\hat{\sigma}_j = \frac{1}{n} \sum [S - E(S_i)]^2.$$

We then use Bayes's method to estimate μ_j. In reality, we may have a priori knowledge of each judge's scoring tendency. Even if not, we still have reason to assume an a priori distribution of each judge's score. If the prior parameters are (μ_0, σ_0^2), then the posterior parameter is

$$\hat{\mu}_j = \frac{n \cdot E(S)}{n + \left(\frac{\hat{\sigma}_j}{\sigma_0}\right)^2} + \frac{\left(\frac{\hat{\sigma}_j}{\sigma_0}\right)^2 \cdot \mu_0}{n + \left(\frac{\hat{\sigma}_j}{\sigma_0}\right)^2}.$$

Then we can use $\text{quantile}(N(\hat{\mu}_j, \hat{\sigma}_j^2), \text{LEVEL})$ as the score threshold, where $1-\text{LEVEL}$ is the expected proportion of papers should be retained. One suitable value is

$$\text{LEVEL} = 1 - \frac{W \cdot J}{P}.$$

Test of the Model

The most important test is to verify that the model makes sense. We do a computer simulation to see how our model behaves as the two practical factors are gradually taken into account. As detailed later, our results agree with our expectations (**Feasibility Test**). In addition, a finding in the course of testing leads us to make some improvement to the model. A more thorough test is made by revising the parameters (**Stability Test**). Lastly, we try to apply our model to more complicated example in real world. The results meet reality very well (**A Real-Life Example**).

Feasibility Test

We fix $P = 100$, $J = 8$, and $W = 3$.

Test of the Basic Model

We assign P papers absolute scores of $1, 2, \ldots, 100$. These values are used only to provide the relative order of the papers.

All papers are randomly allocated to 8 judges at the beginning of the simulation. We calculate the total judging time.

The results of 1,000 iterations of simulation (see **Figure 2**) show that the basic model can select the top three in quite a short time, as we analyzed before.

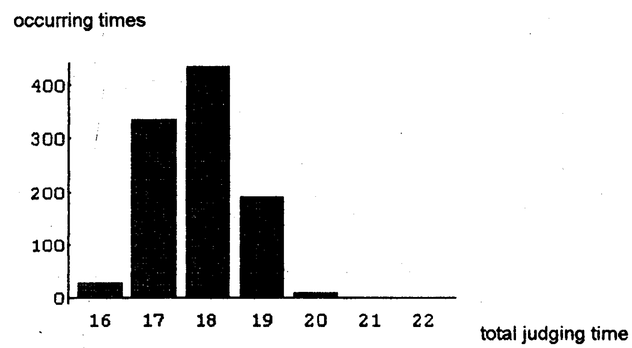


Figure 2. Frequency vs. total judging time.

Take Misjudgment into Account

These two simulations are vital:

- *Simulating the distribution of the absolute score.* Generally, we have reason to use a normal distribution. In order to assign the scores of 100 papers, we generate 100 random numbers following $N(60, 30^2)$, truncated at 0 and 100.

- *Simulating the score given to one paper.* We simulate a judge's scoring by adding a normal random number to the absolute score of the paper.

The quantity σ_1/σ_0 should be fairly small (say, ≤ 0.1), because a judge should have good competence in judgment. We take σ_1/σ_0 = 1/30, 2/30, and 3/30 as cases in our simulation.

We also did a theoretical estimate for these cases of P_{error} under the worst of circumstances. Take $\sigma_1/\sigma_0 = 3/30$ for example. An error occurs when one or more of $P_{(7)}, P_{(8)}, \ldots$ enter the final three. The probability of $P_{(7)}$ entering the final three contributes the most to P_{error}. We let $P_{\text{error}}(i,j)$ be the probability of misjudging papers i and j. We approximate P_{error} by the probability of $P_{(7)}$ entering the final three:

$$\begin{aligned} P_{\text{error}} &\approx \frac{1}{8}P_{\text{error}}(3,7) + \left(\frac{1}{8}\right)^2 \sum P_{\text{error}}(3,i)P_{\text{error}}(i,7) + \cdots \\ &\approx \frac{1}{8}P_{\text{error}}(3,7) + \left(\frac{1}{8}\right)^2 \sum_{i=4}^{6} P_{\text{error}}(3,i)P_{\text{error}}(i,7). \end{aligned}$$

We computed $P_{\text{error}}(i,j)$ using Mathematica by calculating the area of the shaded region in **Figure 1**. In this way, we get the estimate $P_{\text{error}} \approx 0.4\%$.

The results of the simulations accord with the theoretical estimate (see **Table 2**).

Table 2.

Results of 1,000 trials for each value of σ_1/σ_0 vs. theoretical estimates.

$P = 100$, $J = 8$, $W = 3$

σ_1/σ_0	Mean T	Max T	Errors	Observed P_{error}	Estimate of P_{error}
1/30	17.7	21	0	.000	10^{-7}
2/30	17.7	21	0	.000	.0006
3/30	17.8	21	4	.004	.004

An Extra Improvement to the Model

The simulation results demonstrate that the model behaves reasonably so far. Surprisingly, a slight modification improves the model remarkably. If in the first round we select more papers, say W_1 instead of W, and select W papers from the next round, we find that P_{error} declines greatly but the total judging time is scarcely affected. The chances of a excellent paper being rejected in the first round are much more than in the later round, because the papers rejected after the first round are read by only one judge, while those rejected later are read by more judges. **Table 3** gives simulation results for several values of W_1.

Table 3.

Results of 1,000 trials for each value of W_1.

$P = 100$, $J = 8$, $W = 3$

W_1	Mean T	Max T	Errors
6	17.9 ± 0.9	20	0
5	17.8 ± 0.8	20	0
3	17.8 ± 0.8	20	1

Take Judges' Systematic Biases into Account

We might as well simulate the scores from different judges by using the normal distribution with randomly generated mean value and variance.

Using the method offered in **Design of Model**, we get the results of **Table 4**. With increasing LEVEL, the total judging time declines but more errors occur; it is difficult to minimize both time and number of errors.

Table 4.

Results of 1,000 trials for each value of LEVEL.

$P = 100$, $J = 8$, $W = 3$

LEVEL	Mean T	Max T	Errors
50%	22.0	24	0
70%	19.5	23	3
75%	18.7	21	4
80%	18.0	21	4
85%	17.2	20	7
90%	16.3	19	11

Stability Test

We change the parameters P, J, W to test the model's stability. **Table 5** gives the results of 100 iterations for each of several groups of parameters.

Table 5.

Results of 100 trials, for $\sigma_1/\sigma_0 = 3/30$, for each combination of values of P, J, and W.

P	J	W	Mean T	Max T	Errors	$\lceil P/J \rceil + W + 2$
50	4	4	18.1 ± 1.2	22	0	19
80	8	3	14.9 ± 0.9	17	0	15
100	7	3	19.5 ± 0.7	21	0	20
	8	3	17.7 ± 0.8	20	0	18
	8	4	18.8 ± 0.8	21	0	19
		5	19.8 ± 0.9	22	1	20
	10	3	15.4 ± 0.9	18	1	15
120	8	3	19.8 ± 0.9	22	0	20
140	8	3	23.0 ± 1.0	27	1	23
	13	1	14.4 ± 0.7	16	3	14
		2	15.8 ± 0.8	18	1	15
		3	16.9 ± 0.8	19	0	16
		5	18.2 ± 0.9	21	0	18

Analyzing these data, we discover an empirical formula

$$\left\lceil \frac{P}{J} \right\rceil + W + 2,$$

which fits the data for the average value of T wonderfully. Another finding is that small W/P will cause considerable P_{error}. So when W/P is too small (say $\leq 1/100$), the model does not work well. But properly reducing the number of papers rejected in each round will reduce P_{error}.

A Real-Life Example

One idea for testing our model would be to use data from a tennis competition. The table of international standings can be treated as the absolute order, and the result of each formal match acts as a "judge." It is a pity that we have no data!

So we use a substitute for the data of a real competition. We obtained from our department real scores of 50 students for three semesters taking the same three courses. We consider that the sum of each student's three scores stands for the student's level in this major; we take it as an absolute score, and this gives the absolute ordering. In any one semester, the order of students' scores can differ from the absolute order. So we can use these data to simulate a contest, in which each semester acts as a judge assigning a score.

We use these data in our computer program and get the results of **Table 6**.

Table 6.

Results of analysis of departmental data ($P = 50$, $J = 3$).

W	T	papers selected
2	19	$P_{(1)}, P_{(3)}$
3	22	$P_{(1)}, P_{(2)}, P_{(3)}$

Generalization

How to Budget?

Funding for the contest constrains both the number of judges that can be obtained and the amount of time that they can judge.

Assume that each judge can mark n papers/day and the judge's salary is \$$s$/day. We can hypothesize that funding f is a function of T, n, and J: $f = f(T, n, J)$, where T is total judging time. Obviously, $\partial f/\partial J > 0$, $\partial f/\partial T > 0$, $\partial f/\partial n < 0$, and $T = T(J)$. Fortunately, we have the an empirical formula

$$\left\lceil \frac{P}{J} \right\rceil + W + 2.$$

A reasonable functional form for f is

$$f = \left\lceil \frac{T}{n} \right\rceil \cdot J \cdot s = J \cdot s \cdot \left\lceil \frac{\left\lceil \frac{P}{J} \right\rceil + W + 2}{n} \right\rceil .$$

Since $k \leq \lceil k \rceil \leq k + 1$, we get

$$\frac{(P + (W+2) \cdot J) \cdot s}{n} \leq f \leq \frac{(P + (W + 3 + n) \cdot J) \cdot s}{n},$$

which allows us to budget for the contest if the number of the judges has been given (see **Table 7**).

Table 7.

Cost of the contest, for various combinations of papers per day per judge and number of judges.

n	J	min f	max f
15	8	$3,080	$6,067
20	8	$2,310	$5,250
15	7	$2,987	$5,600
20	7	$2,240	$4,812

On the other hand, we can turn the equation around into the form

$$\frac{f \cdot \frac{n}{s} - P}{W + 3 + n} \leq J \leq \frac{f \cdot \frac{n}{s} - P}{W + 2}.$$

According to this, if funding is known, we can decide the number of judges (see **Table 8**). Of course, these are rough estimates.

Table 8.

Number of judges that can be hired, for various combinations of number of papers per day per judge and budget.

$P = 100$, $s = 350$, $W = 3$

n	f	min J	max J
15	$5,000	6	22
10		3	8
15	$7,000	10	40
10		7	20

Applying the Model to Different Kinds of Competition

For contests that give awards to just a few winners, our model is an effective and rational scheme. For contests that give various awards at different levels, we can modify a few parameters in our model. There are two methods.

- Method 1: Suppose that the contest committee expects to classify the participants into different levels in some given proportions, say four levels of 5%, 10%, 35%, and 50%, similar to the MCM. In the first round, we reject 50% as Successful Participation; reject 35% in the 2nd round as Honorable Mention; reject 10% in the third round as Meritorious; the remaining 5% are Outstanding.
- Method 2: We set the value of LEVEL as needed in each round to distinguish participants of different levels. This method is more flexible and fairer than Method 1.

Final Scheme

We summarize our final scheme:

- Divide the judging process into several screening rounds and follow the principles below in each round until W papers remain.
- Use a scoring scheme.
- Do not compare the scores from different judges.
- In the first round, distribute papers to all judges evenly. After scoring, select out the top $2W$ papers in each group to enter the next round.
- At the end of each round, for each paper, calculate the number of papers better than it (which we call the current rank of the paper), then reject every paper whose current rank is more than $W - 1$.
- At the beginning of each round, dispatch papers with a close current rank to the same judge, if possible.
- The number of papers distributed to each judge in each round should be as equal as possible.

Our Suggestions

- Properly reducing the number of rejected papers in the first round would decrease the error probability.
- Altering the number rejected in each round as needed is helpful in competitions that determine different levels of the participants.
- To be more practical and efficient, we suggest prejudging the papers at first, that is, rejecting the papers of distinctly poor quality.

- Between rounds, have some discussion among judges so that they gain some knowledge of the levels of papers as a whole. Such a feedback mechanism surely helps reduce the standard deviation of judgment.

- When there are about $2W$ papers left, all the judges gather to read the remaining papers together, if time permits, to select the top W papers.

Strengths and Weaknesses

Strengths

- We have shown how our model provides an efficient scheme in correct selecting winners. The model was not only tested in a computer simulation but also proved adaptable to real cases.

- Our model is also very stable. All parameters, which we set arbitrarily, can be changed without changing the quality of the model.

- We obtain from our model an empirical formula for T, based on P, J, and W.

- We use Bayesian estimation to take into account the differences among judges.

- Our model is flexible enough to be applied to different kinds of competitions.

Weaknesses

- We are unable to demonstrate that our model is optimal.

- We would like to be able to improve our estimation of the parameters for each judge.

References

Box, G.E.P. and Tiao, G.C. 1973. *Bayesian Inference in Statistical Analysis*. Reading, MA: Addison-Wesley.

Wang, Yihe. 1986. *Introduction to Discrete Mathematics*. Harbin, China: Harbin Institute of Technology Press.

1997: The Discussion Groups Problem

Small group meetings for the discussion of important issues, particularly long-range planning, are gaining popularity. It is believed that large groups discourage productive discussion and that a dominant personality will usually control and direct the discussion. Thus, in corporate board meetings, the board will meet in small groups to discuss issues before meeting as a whole. These smaller groups still run the risk of control by a dominant personality. In an attempt to reduce this danger, it is common to schedule several sessions with a different mix of people in each group.

A meeting of An Tostal Corporation will be attended by 29 board members of which nine are in-house members (i.e., corporate employees). The meeting is to be an all-day affair with three sessions scheduled for the morning and four for the afternoon. Each session will take 45 minutes, beginning on the hour from 9:00 A.M. to 4:00 P.M., with lunch scheduled at noon. Each morning session will consist of six discussion groups with each discussion group led by one of the corporation's six senior officers. None of these officers is a board member. Thus, each senior officer will lead three different discussion groups. The senior officers will not be involved in the afternoon sessions, and each of these sessions will consist of only four different discussion groups.

The president of the corporation wants a list of board-member assignments to discussion groups for each of the seven sessions. The assignments should achieve as much of a mix of the members as possible. The ideal assignment would have each board member with each other board member in a discussion group the same number of times while minimizing common membership of groups for the different sessions. The assignments should also satisfy the following criteria:

1. For the morning sessions, no board member should be in the same senior officer's discussion group twice.

2. No discussion group should contain a disproportionate number of in-house members.

Give a list of assignments for members 1–9 and 10–29 and officers 1–6. Indicate how well the criteria in the previous paragraphs are met. Since it is possible that some board members will cancel at the last minute or that some not scheduled will show up, an algorithm that the secretary could use to adjust the assignments with an hour's notice would be appreciated. It would be ideal if the algorithm could also be used to make assignments for future meetings involving different levels of participation for each type of attendee.

Comments

The Outstanding papers were by teams from East China University of Science and Technology (Shanghai, China), Macalester College, Rose-Hulman Institute of Technology, and the University of Toronto. Their papers, together with commentaries, were published in *The UMAP Journal* 18 (3) (1997): 297–350.

Problem Origin

The problem was contributed by Don Miller (St. Mary's College, IN).

Practitioner's Comments

Vijay Mehrotra (CEO, Onward Inc.) wrote: As an operations management consultant, I am used to dealing with difficult problems, incomplete information, and unclear objectives. My profession requires

- a willingness to wrestle with such assignments by understanding the key business goals and issues;
- a desire to solve the problems by finding the right roles for the right people, models, processes, and information systems;
- and an ability to "sell" our solutions by presenting both our methods and our results clearly to diverse and demanding audiences.

That's why I loved this problem. It is a decidedly nontrivial combinatorial optimization problem with lots of different dimensions. Because it is not a cookie-cutter problem, there is no "right" way to solve it. The problem's complexity prevents teams from using a standard modeling framework and turning the crank.

Accordingly, the best papers tackled the problem with varied methods, including simulated annealing, greedy algorithms, graph theory, and integer programming. These papers explicitly included analysis of all the desired elements (minimal common membership, maximal interaction between different board members, in-house representation, senior officer group restriction).

Another common theme in this year's winning entries was an understanding of the power of good abstraction. Along with a precise and well-presented mathematical formulation of the problem, a top paper described how this abstract problem formulation related to the "real" problem. In turn, the quality of the abstraction is directly related to

- adaptability of the solution to different problems or slightly different conditions, and
- computational feasibility of the selected solution method.

Among the contest papers were several elegant formulations that couldn't be solved and many clever solutions that couldn't be extended. None of those papers was denominated Outstanding.

Finally, each of the winning entries took the time to examine critically the quality of the scheduling solution generated by their modeling methods. It is challenging to define a standard for what a "best" solution is, yet this type of yardstick is essential for assessing how well a specific method works.

Author-Judge's Comments

Problem author and contest judge Don Miller (St. Mary's College, IN) remarked that what makes the situation particularly open-ended is that a "good mix" of board members is not clearly defined. A team must identify some sort of measure of "goodness" of a particular solution. Doing so involves making assumptions, by answering questions such as:

- Is the third meeting of two board members worse than the second?
- Is the second meeting of two in-house members worse than the second meeting of two regular members?
- How should the second meeting of an in-house member with a regular member be evaluated?
- Does increasing the time between sessions in which two members are in the same group reduce the "cost?"
- How does the "cost" of having two board members fail to meet compare to that of having them meet more than once?
- Is common membership in the form of A-B-C worse than common membership of the form A-B and C-D?

The quality and justification of such assumptions were weighted heavily by the judges in their evaluation. Assumptions considered unreasonable include:

- It is better to have a member skip a session than to be in a session with the same member again.
- To minimize common membership, the number of groups for the afternoon sessions may be increased from four to five.
- To ensure that everyone meets at least once, there will be only one group for the seventh session.

The problem statement calls not just for a solution to the An Tostal problem but also for a simple algorithm usable with other parameter values. Papers providing only a "brute force" solution for the An Tostal situation were screened out early.

The most common methods were simulated annealing and the greedy algorithm. However, few teams addressed the generalized problem for future meetings with any or all parameters changed. One team that used the greedy algorithm noted that local optimization—that is, optimization at the session level—does not guarantee global optimization, and thus it may be desirable to allow a second encounter of two members early in the day.

Several teams used a different algorithm foir last-minute changes than for initial assignments. One team doing this made a conscious effort to alter drastically the schedule of a few people rather than modify slightly a greater number of schedules. Another team felt it desirable to consider both balance (of membership of groups) and mixture (pairs that meet a disproportionate number of times); they minimized the geometric mean of these two measures. Yet another team decided that if a pair of members must meet twice, doing so is less "costly" if the time between meetings is maximized.

The four Outstanding papers had many similarities. All decided that it would be desirable to keep the size of the groups for each session as equal as possible. Most observed that 532 pairings would be needed, a value achieved only by the team from East China University of Science and Technology. Some argued effectively that uneven group sizes increase the number of pairings needed. All four teams recognized that 406 pairings are necessary if each board member were to meet each other board member. In fact, all these teams reported the number of times each of these pairings (handshakes) occurred in their final or "best" solution, as shown in **Table 1**.

Table 1.

Occurrences of numbers of pairs in final solutions.

Team	Number of pairs			
	0	1	2	3
East China Univ. of Science and Tech.	26	253	102	25
Macalester College	40	214	138	14
Rose Hulman Institute of Technology	32	218	152	4
University of Toronto	33	226	134	14

The team from Rose-Hulman Institute of Technology got only four groups meeting three times, since their penalty for this situation was modeled as powers of four. The team from Macalester College had the largest number of pairs that never met, 40; they were the only team of the four that looked beyond individual pairs in developing their objective function. For their solution, "No two discussion groups have more than two members in common."

Three of the four Outstanding papers used a form of the greedy algorithm. The other, from Macalester College, stood out for its explanation of how to use simulated annealing. Its strong objective function was the sum of four objectives, with a penalty for more than one repeat pairing in a group. In addition, the paper contained a nice proof that if no two pairs of board members are together more than two times, then some pair is never together.

The paper from the Rose-Hulman team stood out for its writing and for its statistical comparison of schedules. Schedules made randomly by the greedy algorithm and by a modified greedy algorithm were used as a baseline for other methods.

Finally, the University of Toronto team provided excellent proofs on bounds for solutions, which they applied to the An Tostal situation.

Note: The name “An Tostal” has no real meaning as far as the problem is concerned. It was the name of the spring-quarter weekend of celebration just before final exams at Kent State University, where Prof. Miller was an undergraduate.

Using Simulated Annealing to Solve the Discussion Groups Problem

David Castro
John Renze
Nicholas Weininger
Macalester College
St. Paul, MN 55105

Advisor: Karla V. Ballman

Introduction

Our task is to assign 29 corporate board members to a sequence of discussion groups organized into seven sessions, of which three are morning sessions led by senior officers and four are afternoon sessions not led by senior officers. We wish to find a combination of group assignments that best satisfies the following objectives:

- In the morning sessions, no board member should be in the same senior officer's discussion group twice.
- No group should have too many in-house members (there are nine in-house corporate employees among the 29 members).
- The number of times any two board members are in the same group should vary as little as possible.
- No two groups should have a large number of common members.

The specific case given assumes that all members will participate in all sessions and that there will be six discussion groups for each morning session and four for each afternoon session. Our model ought to be capable of producing good answers quickly under these assumptions and of adjusting to more general configurations of board members. In particular, it should be able to adjust to small changes, such as individual additions or subtractions of board members, without recalculating all assignments from scratch.

Model Design Considerations

This problem is essentially one of optimization. The different potential assignments of board members form a solution space, and we must optimize the “fitness” of our assignments (how well they satisfy our objectives) over that solution space. Thus we have to consider the issues that come up when designing any optimizer: time and space efficiency, flexibility, and optimality of the solution. We also have concerns particular to this problem.

Multiple Objective Satisfaction

Most optimization problems involve maximizing a single objective function; here we have four objectives. The problem statement doesn’t tell us whether, for example, to consider minimizing common group membership more important than minimizing the number of times any two board members meet. Presumably, we want to use some sort of weighted combination of the objectives as the function to optimize over the solution space—but determining how to combine and weight them is nontrivial.

Large Solution Space

The number of ways of assigning the board members to the discussion groups is enormous. Since each board member goes into seven sessions, we have a total of $29 \times 7 = 203$ variables; since there are six possible assignments for each member in the morning sessions and four for the afternoon sessions, the total number of possible solutions is on the order of $6^{87} \times 4^{116} \approx 3 \times 10^{137}$. Furthermore, no matter how we define our objective function, the solution space will probably contain a large number of local optima. This has two implications. First, it’s probably impossible to find a global optimum (and it isn’t absolutely necessary here). Second, we need a solution model that doesn’t require searching over a significant fraction of the solution space.

Fast Readjustment

Since board members may drop out or in at the last minute, we want a way of taking a precomputed solution and adjusting it for small changes in the member configuration. This adjustment should be significantly faster than a complete recomputation and should also involve fewer changes of assignment.

Simulated Annealing

We chose simulated annealing as the basis for our model, as it offers the best chance of constructing an effective solution-finding algorithm.

The Simulated Annealing Process

The simulated annealing algorithm can be described as follows:

1. Start with an objective function that is evaluable for every point in the solution space, a randomly chosen point in that space, and an initial "temperature" value.
2. Evaluate the initial point's objective function value.
3. Perturb the point by a small amount in a random direction.
4. Calculate the objective function value of the perturbed point.
5. If the new function value is better than the old one, accept it automatically, staying at the perturbed point.
6. If the new function value is worse, decide probabilistically whether to stay at the perturbed point or go back to the old one. The probability of not staying at the new point should depend on how low the temperature is and how much worse the new point is than the old one.
7. Lower the temperature slightly and go to step 3. Iterate until the temperature is close to 0 or the solution stops changing.

The algorithm essentially does a hillclimb through the solution space, with occasional random steps in the wrong direction. The temperature controls how likely the algorithm is to take a step the wrong way; at the beginning, the algorithm jumps almost completely randomly around the solution space, but by the end it almost never takes a wrong step. The idea is that the random steps allow the hillclimber to avoid getting stuck at local minima or maxima. In most implementations of simulated annealing, the probability of acceptance of a wrong-way step is $e^{-d/T}$, where d is the difference between the old and new objective functions and T is the temperature. The temperature typically starts at a value sufficient to give almost any degree of wrong-way movement a significant probability of acceptance, and decreases exponentially from there.

The original idea for simulated annealing comes from statistical mechanics. The molecules of a liquid move randomly and freely at high temperatures; when the liquid is cooled and then frozen, the molecules essentially "search" for the lowest energy state possible. The minimum energy state occurs when the molecules are arranged in a specific crystalline structure. If you cool the liquid slowly, the molecules will have time to redistribute themselves and find this structure; if you cool it quickly, they will typically get "stuck" at a higher-energy, noncrystalline state.

Reasons to Use Simulated Annealing

A number of factors influenced our decision to apply simulated annealing to our problem:

- It's fast. Simulated annealing requires evaluation of the objective function over a relatively small number of points to achieve its result.
- It's simple. Everything except the fitness function evaluation can be done in well under 100 lines of C code. It isn't difficult to understand or debug.
- It lends itself well to discrete solution spaces and nondifferentiable objective functions. As long as you can provide a random jump between solution points and a way of evaluating the function at any point, it can work; many other methods require a continuous solution space, or demand that you evaluate the function's derivative.
- It has a track record of success. Simulated annealing has been used for applications as diverse as stellar spectrum analysis and chromosome research.
- We were all well acquainted with simulated annealing, and one of us had previously used it to solve a similar partitioning problem.

Alternatives

Heuristic Algorithms

One could try to get a solution by using some sort of intuitive rules about how to rearrange things, much as a human secretary would. This would likely be too slow for a large number of members, though, and coming up with good intuitive rules that a computer can implement is difficult.

Gradient Methods

Traditional gradient descent methods are another possibility. These, however, generally require that the objective function's derivative be evaluated or at least estimated, and our objective function is extremely difficult to express mathematically. Furthermore, they run the risk of getting stuck at local minima.

Integer Programming

Integer programming is commonly used for discrete optimization problems like this one. But none of us had experience implementing it, and we feared that the large size of the solution space might make it too slow.

Genetic Algorithms

We could establish a pool of potential solutions that would "evolve" toward an optimal solution through a process analogous to natural selection. But this, too, would likely be too slow and complex for our problem.

Modeling the Solution

We designed and coded in C a program that uses simulated annealing to solve a general set of discussion-group problems, including the one given.

The Data Structure

Our approach to the problem began with data structure design: We needed a way to encode the whole list of assignments that would allow us to code the annealing process straightforwardly, and preserve important testing and organization data for analysis of the annealing results.

The data structure `partition` has three levels of organization:

1. The top level, `partition`. This contains a lot of data pertaining to the whole solution structure: the number of morning and afternoon sessions, the number of groups in each type of session, and the total number of members and of in-house members participating in each session. It also contains some data relevant to the objectives concerning pairs of members and common membership (more on that below).
2. The list of groups contained in the `partition`. Each `group` corresponds to one of the discussion groups in one of the sessions and has a size variable.
3. The list of people contained in each `group`. Each `person` has a name, a number and a flag indicating whether they're in-house. Thus each instance of `person` corresponds to one member's participation in one session. Note that this allows members to participate in only some of the sessions.

The Objective Function

We chose as the objective function a weighted sum of four subfunctions. Each subfunction takes a partition and calculates how close it comes to satisfying one of the objectives given in the problem. The results are all expressed in terms of "badness," or the degree to which the partition fails to satisfy an objective; thus our problem becomes the minimization of the "badness function."

Senior officer nonrepetition

The first objective we consider is that no board member should be in the same senior officer's morning group twice. Our first subfunction simply takes the morning groups headed by each senior officer and checks for repetitions in their member lists. The total number of repetitions—that is, the total number of times a board member is in the same senior officer's morning group more than once—is multiplied by one weight to form the first part of the badness function. Since we want zero repetitions, and want the annealer to stay with solutions with zero repetitions once it finds them, we also have a “zero bonus” that subtracts a given amount from the badness if there are zero repetitions.

In-house members

The second objective states that no group should have a disproportionate number of in-house members. Our second subfunction calculates the amount of disproportionality in the distribution of in-house members among groups. For each session, it uses the number of total members and in-house members participating in the session to compute ideal floor and ceiling values for the number of in-house members in each group. (If the number of in-house members is a multiple of the number of groups in the session, then the floor and ceiling are the same; otherwise they differ by one.) Then it goes through each group in the session and tests to see if the number of in-house members is more than the floor or less than the ceiling; if so, it adds to the disproportionality sum the difference between the floor or ceiling and the actual number. So, if the ideal ceiling were two in-house members per group, a group with four in-house members in it would add two to the disproportionality sum.

We want the disproportionality sum, like the repetition sum, to be zero, and so implement a zero bonus; we set this equal to the disproportionality weight rather than making it separately adjustable, for reasons discussed below.

Pairings of board members

Our third objective is to try to ensure that each board member meets each other board member approximately the same number of times. Our third subfunction does this by computing ideal floor and ceiling values for that meeting number, just as the second does for the proportion of in-house members. In the `partition` structure we keep a matrix whose elements correspond to the number of times each pair of board members are in the same group. The third subfunction recalculates this matrix by looking at each pairing in each discussion group; derives the ideal floor and ceiling values from the matrix; and then goes through the matrix again, adding up the instances in which the number of times a pair of members meets is less than the floor or more than the ceiling. This sum becomes the *pairwise anomaly count*.

We also calculate the maximum number of times any two board members meet, and use that as well as the pairwise anomaly count in the badness calculation. The reasoning behind this is not only because it’s desirable for most pairs of members to meet the same number of times, but also to ensure that no pair of members is in a hugely disproportionate number of groups together. We don’t want a solution with one pair that is in every group together but with no other anomalous pairs to be preferable to a solution with many slightly anomalous pairs but with no hugely anomalous ones.

Here, we have no zero bonus for the anomaly sum, because the meeting criterion ought to be satisfied “as much as possible,” rather than absolutely, and because it’s impossible in many cases (including ours) to drive it to zero.

Common membership

Our final objective is to ensure that no two groups have a disproportionate number of common members. The fourth subfunction is almost precisely analogous to the third. Again, we have a matrix in the `partition` structure, listing for each pair of groups how many members they have in common; again, we calculate this matrix, use it to derive an ideal mean, and find the sum of deviations from that mean and the maximal deviation. Here, however, we don't count as deviant groups those that have fewer than the mean number of common members. It doesn't matter if two groups have no members in common, only if they have too many.

The Annealing Iteration Process

Once we have the "badness function" described above, we perform simulated annealing. The relevant implementation details are:

- The perturbation: A single swap of two members is the unit of random perturbation. Our program randomly chooses a session, two discussion groups within that session, and a member in each group, and then swaps them.

- The initial configuration: The file-reader dumps the members from the file into the `partition` in order, giving an extremely bad configuration. Our program does random swaps on that configuration for a random number of times (between 1 and 32,767), producing a "shuffled" initial configuration.

- The starting temperature setting and the rate of exponential decay: These are two key variables that determine how long the annealing takes and what the temperature profile is.

  For the starting temperature, we just use the badness value of the starting configuration; this means that at the starting temperature, a solution as bad as our starting one would have a 10% chance of being accepted as a step away from a perfect (zero-badness) configuration.

  For the decay, we used a variety of different rates. Multiplying the temperature by 0.998 at the end of each iteration tends to give the best time/accuracy tradeoff. Slower decay makes for long annealing runs that don't get much better; faster decay doesn't give the process enough room to "jump around" randomly at the beginning, resulting in much worse solutions.

Setting the Weights

We anneal in an attempt to minimize the following badness function:

$$\begin{aligned}\texttt{badness(partition)} &= w_0 * \texttt{reptotal} - w_1 * \texttt{repzero} + w_2 * \texttt{disprop} \\ &\quad - w_2 * \texttt{diszero} + w_3 * \texttt{pairanom} + w_4 * \texttt{maxpair} \\ &\quad + w_5 * \texttt{commanom} + w_6 * \texttt{maxcomm}\end{aligned}$$

where

- `reptotal` is the number of times a member is in the same senior officer's group twice;
- `disprop` is the disproportionality sum for in-house members;
- both `repzero` and `diszero` are 1 if `reptotal` and `disprop` both are 0, and both are 0 otherwise;
- `pairanom` and `maxpair` are the anomaly score for pairwise meetings and the maximum number of times a pair meets;
- `commanom` and `maxcomm` are the anomaly score for group common membership and the maximum number of members a group has in common; and
- $w_0, \ldots, w_6$ are integer weights.

How should we set the weights? We did so by trial and error. The set of weights $w_0 = w_1 = 1200$, $w_2 = 1000$, $w_3 = 400$, $w_4 = 4000$, $w_5 = 100$, and $w_6 = 500$ produces extremely good results for a 9,000-iteration annealing run on the standard problem configuration. Such a run takes about 5 min on an HP 712/60 workstation. Using these weights, we could repeatedly produce solutions that had no senior officer repetitions, no instances of in-house member disproportion, no pair of members that met more than three times, and no pair of groups with more than three members in common.

We used somewhat different weights for a longer annealing run to produce our very best solution, and also adjusted the weights to produce solutions for different session configurations, as we will describe below. But the set of defaults above appeared to work remarkably well for a variety of configurations.

Some things to note about the default weights:

- The zero bonus for senior officer nonrepetition equals the minimization weight. That works so well that we hard-coded in the same equality for in-house disproportionality, rather than making another adjustable zero bonus.
- The ratio of pairwise minimization weight to pairwise maximum weight is 1 to 10. That means that the algorithm considers reducing the pairwise anomaly score by 10 equivalent to reducing the pairwise maximum by 1. Lower ratios tend not to force the pairwise maximum low enough; higher ratios tend to prevent the annealer from occasionally making the pairwise maximum higher by 1 on its way to a much lower pairwise anomaly score.

- The common membership weights are quite small. We found that good common membership configurations were strongly correlated with good pairwise meeting configurations; that is, configurations in which no pair of members met an inordinate number of times also tended to be configurations in which no two groups had too many common members.

Our Solution

Using extra-long runs and adjusting the weights by trial and error, we eventually produced the solution in **Tables 1–2**. The in-house members are 1–9 and the non-in-house members are 10–29.

Table 1.
Assignments by discussion group.

Session	Group	Members
Morning 1	1	27 22 20 17 1
	2	4 28 21 3 13
	3	10 11 8 29 18
	4	14 19 24 7 23
	5	15 9 5 16 12
	6	6 25 2 26
Morning 2	1	13 29 2 6 28
	2	10 16 9 19 25
	3	7 23 20 22 12
	4	5 11 26 17 3
	5	8 14 20 27 4
	6	24 21 1 15
Morning 3	1	16 8 19 21 5
	2	2 18 22 1 23
	3	15 26 17 9 14
	4	12 25 4 29 27
	5	2 13 20 11 6
	6	28 7 3 10
Afternoon 1	1	27 26 24 2 8 10
	2	6 3 22 15 19 4 18
	3	11 16 23 5 25 14 1 28
	4	21 9 13 17 29 7 12
Afternoon 2	1	23 6 5 21 10 27 12
	2	3 15 29 14 2 25 20
	3	26 9 1 8 28 19 22 13
	4	7 24 4 17 16 11 18
Afternoon 3	1	15 4 1 13 2 10 16
	2	5 24 12 25 22 17 8
	3	21 28 14 20 7 26 18 6
	4	27 23 11 19 3 9 29
Afternoon 4	1	7 25 27 13 19 5 18
	2	21 14 22 11 10 2 9
	3	12 29 3 6 16 24 26 1
	4	4 23 17 15 28 20 8

Table 2.

Assignments by board member.

	Morning 1	Morning 2	Morning 3	Afternoon 1	Afternoon 2	Afternoon 3	Afternoon 4
In-house members							
1	1	6	2	3	3	1	3
2	6	1	2	1	2	1	2
3	2	4	6	2	2	4	3
4	2	5	4	2	4	1	4
5	5	4	1	3	1	2	1
6	6	1	5	2	1	3	3
7	4	3	6	4	4	3	1
8	3	5	1	1	3	2	4
9	5	2	3	4	3	4	2
Other members							
10	3	2	6	1	1	1	2
11	3	4	5	3	4	4	2
12	5	3	4	4	1	2	3
13	2	1	5	4	3	1	1
14	4	5	3	3	2	3	2
15	5	6	3	2	2	1	4
16	5	2	1	3	4	1	3
17	1	4	3	4	4	2	4
18	3	5	2	2	4	3	1
19	4	2	1	2	3	4	1
20	1	3	5	1	2	3	4
21	2	6	1	4	1	3	2
22	1	3	2	2	3	2	2
23	4	3	2	3	1	4	4
24	4	6	5	1	4	2	3
25	6	2	4	3	2	2	1
26	6	4	3	1	3	3	3
27	1	5	4	1	1	4	1
28	2	1	6	3	3	3	4
29	3	1	4	4	2	4	3

How Good Is This?

In this configuration, no member is ever in the same senior officer's morning discussion group twice. No group contains a disproportionate number of in-house members (the morning groups contain 1 or 2, the afternoon ones 2 or 3).

No pairs of members are in the same group together more than 3 times. Of the possible pairs of members, 40 never meet; 214 meet once; 138 meet twice; and 14 meet three times. Thus the pairwise anomaly score is 54, and the vast majority of members meet one another a "reasonable" number of times (the mean number of meetings is about 1.3). It is possible to achieve a configuration in which no two members meet more than twice, but not while preserving the other objectives.

Also, no two discussion groups have more than two members in common. This is as low as possible.

We conclude that this configuration satisfies all the objectives well and three of the four perfectly; it is likely extremely close to the global optimum.

Typical Results on the Standard Configuration

A typical annealing run, with default weights, will zero out the senior officer repetition and in-house disproportionality. It will also reduce the maxima of member-pair meetings and group common membership to 3. It will not usually reduce maximum group common membership to 2, and it typically gives a pairwise anomaly score of 50 to 60.

Thus, the standard annealing run's result isn't quite as good as our best; but it's nearly as good and much easier to achieve. It considers only about 9,000 different sets of assignments in reaching its solution, and takes about 6 min.

Generalizing the Model

More and Different Board Members

We tested the annealing process on data sets with a variety of different numbers of board members, ranging from 20 to 100. We also tried keeping the number of members at 29 and adjusting the proportion of in-house members. Finally, we tried keeping the existing set of 9 in-house and 20 other members but changed the number of senior officers, the number of morning and afternoon sessions, and the number of groups in each afternoon session. In all cases, we annealed with the same default weights used for the standard configuration.

The model responded extremely well in each case. **Table 3** shows the times required for a full annealing run on the standard session configuration with various numbers of members.

Table 3.
Run-time results.

Members			Run-time
In-house	Others	Total	
5	10	15	2:56
9	20	29	6:38
12	27	39	8:25
15	33	48	9:53
18	41	59	22:06
30	70	100	58:01

We also tried changing the profile to 4 in-house and 25 other members, and to 14 in-house and 15 other members. In both cases, the annealing still drove the in-house disproportionality to zero and reduced the pairwise and commonality maxima to 3.

We tried a total of five different changes in the session / group profile; these ranged from decreasing the number of groups in each morning session to two to having only one morning session and six afternoon sessions. In almost all cases, the annealing produced solutions that were as good as could be expected, given the limitations of the sessions and groups.

The only exception occurred when we tried having three officer-led groups for each of the three sessions. Then the default weights failed to minimize the senior officer nonrepetition criterion, which is much harder to achieve with three groups than with six. Increasing w_0 to 2,000 and rerunning gave much better results.

Different Levels of Session Participation

The data structure and annealing code are designed to deal transparently with members who attend some but not all of the sessions, by considering each member's participation in each session as a separate variable. We tested our code on several different session-participation variations, including:

- making the in-house members not attend the afternoon sessions,
- making some of the non-in-house members not attend the morning sessions, and
- introducing new members who go only to one morning session.

In all cases, the annealing produced good results—zero senior officer repetitions, no more than two instances of in-house disproportionality, maximal pairwise meetings, and group common memberships at one more than the ideal mean.

Adjusting Quickly to Last-Minute Changes

Another important consideration is how well our model can deal with small changes in the configuration of board members—one or two new board members added or deleted, say. We investigated two approaches to adjusting an already-annealed configuration, reannealing and flip-path search. Both are motivated by the idea that the new configuration's best solution should be quite close to the old one's. We tested these approaches on several small modifications: single and double additions and deletions to all sessions, plus additions to only one or two sessions.

Reannealing

Since we want to stay in the neighborhood of the old configuration and to take less time than the original annealing, we use a much lower starting temperature. Experimentation shows that dividing the regular starting temperature by 1,000 works best. The new configuration often is very different from the old one. This would be undesirable in real-world applications, where you might want to make as few changes to group assignments as possible.

Greedy path search

The alternative is to try to reduce the badness function with as few changes as possible. One way is to try all possible single flips in the configuration, then try all possible combinations of two possible single flips, and so on, always looking for the flip sequence that produces the lowest badness function value.

This approach quickly becomes too slow. There are 2,310 possible flips in the standard configuration, and evaluating all of them takes more than 1 min on an HP 712/60; thus, evaluating all two-flip sequences would take at least 20 h. An alternative, much faster approach is to try all single flips, take the one that produces the lowest badness function, perform that flip, try all single flips from the resulting flipped configuration, and so on. We call this a *greedy path strategy*, and in our tests it brought the new configuration's badness function down acceptably close to the old one's in seven to ten flips.

Furthermore, we observed during our tests of the greedy path search that it tended to do best after doing one flip involving groups in each of the seven sessions. This was especially true when we added a new member to all of the sessions. This makes sense, because it ought to take one flip to put a new member in the "right" place in each session.

So we tried the following variation: Perform the best of the possible flips involving groups in the first session, then perform the best of the flips involving the second session, and so on, to the last session. This requires considering all of the 2,310 possible flips only once. This does not work nearly as well as the original greedy search, probably because this approach fixes the order in which the best flips in the sessions are taken.

We finally found a workable hybrid approach. This approach starts out by finding and taking the best flip from all the sessions, then takes the best flip in each session in order, and finally again finds and takes the overall best flip. It runs in roughly 5 min for the standard configuration and gives test results about as good as for the original greedy path search. It doesn't always work as well as reannealing, but it never requires more than nine (in the standard configuration) single-flip changes.

Improving the Model

Complexity

Our algorithm runs in time approximately proportional to the square of the number n of board members. The pairwise part of the badness calculator goes through a matrix including all $n(n-1)/2$ pairs of members. The run-time is also quadratic in the number of groups, because of the pairs-of-groups common membership matrix.

It would be nice to develop a badness function that runs in linear time and produces a good approximation to our original function value. We could also make smaller efficiency improvements, such as developing a way to update efficiently the pairwise and commonality matrices with each single flip instead of recalculating them on each iteration. Time considerations prevented us from doing this (we tried doing it with the pairwise matrix but never got it to run significantly faster than straight recalculation). We believe that our algorithm runs remarkably well as it is; finding a near-minimum over a solution space of 10^{137} points in 6 min is no small task.

Flexibility

The present model allows only two kinds of group sessions—morning and afternoon—and assumes that all sessions have the same number of groups of the same size (or differing by only one). It wouldn't be too difficult to extend the `partition` structure to allow for a more complex session structure, such as one that would allow for different numbers of senior officers at each morning session.

We could also add different types of board members and specify new objectives based on them. Perhaps, for example, one might want to stipulate that some board members are new, and that new board members should all be in the same discussion groups so that they can get to know each other (or at least that every new member should meet every other member once).

Finally, we could try to devise a general method for setting annealing weights, given a configuration. The default standard weights appear to work well for a large range of configurations, but they are almost certainly not optimal for all configurations.

References

Kirkpatrick, S., C.D. Gelatt, and M.P. Vecchi. 1983. Optimization by simulated annealing. *Science* 220 (13 May 1983): 671–680.

Press, William, et al. 1992. *Numerical Recipes in C.* 2nd ed. New York: Cambridge University Press.

1998: The Grade Inflation Problem

Background

Some college administrators are concerned about the grading at A Better Class (ABC) College. On average, the faculty at ABC have been giving out high grades (the average grade now given out is an A−), and it is impossible to distinguish between the good and the mediocre students. The terms of a very generous scholarship only allow the top 10% of the students to be funded, so a class ranking is required.

The dean had the thought of comparing each student to the other students in each class, and using this information to build up a ranking. For example, if a student obtains an A in a class in which all students obtain an A, then this student is only "average" in this class. On the other hand, if a student obtains the only A in a class, then that student is clearly "above average." Combining information from several classes might allow students to be placed in deciles (top 10%, next 10%, etc.) across the college.

Problem

Assuming that the grades given out are (A+, A, A−, B+, . . .), can the dean's idea be made to work?

Assuming that the grades given out are only (A, B, C, . . .), can the dean's idea be made to work?

Can any other schemes produce a desired ranking?

A concern is that the grade in a single class could change many students' deciles. Is this possible?

Data Sets

Teams should design data sets to test and demonstrate their algorithms. Teams should characterize data sets that limit the effectiveness of their algorithms.

Comments

The Outstanding papers were by teams from Duke University, Harvey Mudd College, and Stetson University. Their papers, together with commentaries, were published in *The UMAP Journal* 19 (3) (1998): 279–336.

Problem Origin

The problem was contributed by Daniel Zwillinger (Zwillinger & Associates).

Practitioner's Comments

According to Valen E. Johnson (Duke University)[1], mentioned that the solutions in the Outstanding papers span the range of previously proposed alternatives to GPAs in terms of model complexity, ranging from relatively simple to highly complex.

Apart from the role of GPA in scholarship allocation, selection to honorary societies, success in the job market, and admission to professional or graduate school, a more subtle influence is its impact on course selection. Because GPA is perceived to play a critical role in a student's career, students commonly select courses based on expected grade. Fewer "hard" courses are taken, and probably fewer courses in science and mathematics. Also, grades spiral up: Students gravitate to courses graded leniently, and professors soften grading standards to ensure adequate enrollments and favorable evaluations by students.

Changing the way GPA is computed can solve all of these problems, but implementing such a change is a difficult proposition. Any change will be opposed by faculty and students who do not benefit from the change. Any modification must be logically consistent, fair, and understandable by non-statisticians. Compromises in these three criteria are inevitable.

The Stetson University team proposed to standardize grades in each class using either the median or mean grade. But for data taking on only two or three values, the median can be very uninformative. When grades are inflated, the median grade is likely to be A or A$-$, but this says little about their relative proportions and less still about proportions of other grades. A compromise between the median and the mean would be a *trimmed mean*, which offers some robustness against outliers while maintaining good statistical efficiency. For example, to to compute the 10% trimmed mean is the mean of the data remaining when the lowest and the highest 5% of grades are ignored.

In my opinion, such standardization compromises fairness because it does not account for the quality of students within a class. Standardization would encourage students to opt out of courses known to be populated by top students.

Consistency is also a problem. To see why, consider two students who take identical courses through their senior years, and receive A$+$ in all of their courses. In the last semester of their senior year, the second student develops an interest in art history and takes an introductory course in that subject (in addition to the other courses that both he and the first student take). Both students again receive A+ in all of their courses, but unfortunately everyone

[1]When he prepared his comments, Prof. Johnson was blinded to the identities of the teams, which were inserted later in editing, and hence was not aware that an Outstanding paper was from a Duke University team.

in the art history course also receives an A+. Which student graduates on top? According to the Stetson method, the first student beats out the second student for valedictorian, even though the second student tied the first student in all of the courses they took together and got an A+ in one additional course. Why? The standardized grade for an A+ in the art history course is 0, which when averaged in lowers the adjusted GPA.

The team from Duke University proposed a regression-type adjustment, in which the grades from each class are adjusted for the difference between the mean grade in the class and the mean adjusted GPA of students in that class. This difference is then used to compute new adjusted GPAs, which leads to new adjustments to the class grades. The team's least-squares estimate of the adjusted GPA is based on the assumption that students tend to receive higher grades in classes taken in their academic majors.

Both adjustment schemes are quite similar to proposals of Caulkin et al. [1996], Strenta and Elliot [1987], Goldman and Widawski [1976], and Goldman et al. [1974]. The model of Caulkin et al. [1996] has the form

$$\text{Grade}_{ij} = \text{True GPA}_i + \text{Course effect}_j + \epsilon_{ij},$$

so that student$i$'s grade in course j is an additive function of their "true GPA" plus a course effect plus a (normally distributed) random error. This approach implicitly accounts for both the grading policies of individual instructors and the quality of students within each class.

The primary drawback of these regression-type models is the assumption that grades are intervally scaled. It usually does not make sense to assume that the difference between an A and A$-$ is the same as the difference between a C and C$-$, or that a D plus a B equal two Cs. Typical grading scales assign more As and Bs than Cs and Ds, and by not taking the ordinal nature of grade data into account, substantial statistical efficiency is lost. These models also suffer from the paradox presented above for standardized GPAs. The art history course would lower the second student's "true GPA," though an independent study course would leave it unaffected.

Both the Duke team and the Harvey Mudd team assume that it is possible to assign a single ability score to each student. This assumption is not necessary or appropriate. Each student's true GPA can instead be interpreted as the ability score for the student *in courses that that student chose to take.* The model of Caulkin et al. [1996] and the Duke team variation can be applied without difficulty to colleges in which, say, humanities students and engineering students have no common classes. The grades of engineering students in engineering classes should not be used to estimate their abilities in humanities classes.

It is important to distinguish difficult courses from courses that are graded stringently. They are not (always) the same, so it does not follow that students should be penalized for taking courses that are graded leniently. In fact, a student who receives the highest grade in all classes that she takes should be awarded a high adjusted GPA.

The proposal from Harvey Mudd College is surprisingly close to a statistical model called the *Graded Response Model* (GRM). The basic assumption is that instructors choose thresholds on an underlying achievement scale and assign grades based on the grade intervals into which achievement is observed to fall. Details of this model are discussed in Johnson [1997].

A number of assumptions made by the Harvey Mudd team are not necessary, e.g., that one letter grade corresponds to one standard deviation in student achievement, or that professors do not grade on curves or compare performances of students within classes, or that professors uniformly adjust for course difficulty. The assumptions that

- students select courses randomly,
- students do not gravitate to courses that are graded leniently, and
- professors have accurate perceptions of student achievements

are questionable and are not required for the GRM; they could be eliminated.

The Harvey Mudd College model handles the two-student paradox mentioned above. Their model also attempts to combine information about the grading patterns of instructors across classes, an aspect of model fitting not normally included even in GRMs. The primary disadvantage of their proposal is its complexity—it is clearly the most difficult model to explain.

Author-Judge's Comments

Daniel Zwillinger remarks that averaging grades results in systematic biases against students enrolled in more rigorous curricula and / or taking more courses. The problem of finding a "better" ranking has no simple "solution." Johnson [1997] refers to many studies of this topic and suggests a technique that was considered—but not accepted—by the faculty at Duke University.

The problem statement suggested that relative rankings of students within courses should be used to evaluate student performance. With this assumption, possible approaches include:

- using relative ranking *with* grade information

  (A useful additional assumption might be that faculty would give grades based on an absolute concept of what constitutes mastery of a course.)

- using relative ranking *without* grade information

In the latter approach (chosen by most teams), an instructor who assigns As to all students in a course provides exactly the same information as an instructor who assigns all Cs to the same students in another course.

Specific items that the judges looked for in the papers included:

- Reference to ranking problems in other fields that use relative performance results, such as chess and golf.

- A detailed worked-out example, illustrating the method(s) proposed, even if there were only 4 students in the example.

- Computational results (when appropriate) and proper consideration of large datasets. Teams that used only a small sample in their computational analysis (say 20 students) did not appreciate many of the difficulties with implementing a grade adjustment technique.

- Mention (if not use) of the fact that even though the GPA may not be as "good" a discriminator as the various solutions obtained by the teams, it seems reasonable that there be some correlation between the two.

- A response indicating understanding of the question about the changing of an individual student's grade. Such a grade change could affect that student's ranking; but if it affected many other students' ranks then the model is probably unstable.

- A clear, concise, complete, and meaningful list of assumptions. Needed assumptions included:

  - The average grade was A$-$, as asserted in the problem statement—amazingly, several teams assumed other starting averages!
  - in an $\{$A$+$, A, A$-$, ... $\}$ system, not *all* grades were A$-$. (Otherwise, there is no hope for distinguishing student performance.)

  Some teams made assumptions that were not used in their solution, were naive, or needed further justification:

  - "Grades form a continuous distribution." As an approximation of a discrete distribution, this is fine. However, several teams allowed grades higher than A$+$, and other teams neglected to convert to a discrete grade when actually simulating grades.
  - "Instructors routinely turn in a percentage score or course ranking with each letter grade." This would be very useful information but is not realistic.
  - "Low grades in a course imply that the course is difficult." A course could be scheduled only for students who at "at risk." Likewise, a listing of faculty grading does not necessarily allow "tough" graders to be identified: An instructor may teach only "at risk" students.

The most straightforward approaches to solving this problem were use of information about how a specific student in a course compared to ...

- ... the statistics of a course. For example, "Student 1's grade was 1.2 standard deviations above the mean, Student 2's grade was equal to the mean, ..." The numbers $\{1.2,\ 0,\ \ldots\}$ can be used to construct a ranking.

- ... other specific students. For example, “in Course 1, Student 1 was better than Student 2, Student 1 was better than Student 3, ...” This information can be used to construct a ranking.

References

Caulkin, J., P. Larkey, and J. Wei, J. 1996. Adjusting GPA to reflect course difficulty. Working paper, Heinz School of Public Policy and Management, Carnegie Mellon University.

Goldman, R., D. Schmidt, B. Hewitt, and R. Fisher, R. 1974. Grading practices in different major fields. *American Education Research Journal* 11: 343–357.

Goldman, R., and Widawski, M. 1976. A within-subjects technique for comparing college grading standards: implications in the validity of the evaluation of college achievement. *Educational and Psychological Measurement* 36: 381–390.

Johnson, Valen E. 1997. An alternative to traditional GPA for evaluating student performance. *Statistical Science* 12 (4): 251–278.

________. 2003. *Grade Inflation: A Crisis in College Education*. New York: Springer.

Larkey, P., and J. Caulkins. 1982. Incentives to fail. Working Paper 92–51. Pittsburgh, PA: Heinz School of Public Policy and Management, Carnegie Mellon University.

Strenta, A., and R. Elliott. 1987. Differential grading standards revisited. *Journal of Educational Measurement* 24: 281–291.

A Case for Stricter Grading

Aaron F. Archer
Andrew D. Hutchings
Brian Johnson
Harvey Mudd College
Claremont, California 91711
{aarcher,ahutchin,bjohnson}@hmc.edu

Advisor: Michael Moody

Abstract

We develop a ranking method that corrects the student's grades to take into account the harshness or leniency of the instructor's grading tendencies.

We simulate grade assignment to a student based on the student's inherent ability to perform well, the student's specific aptitude for the course, the difficulty of the course, and the harshness or leniency of the instructor's grading.

We assume that we have access to a instructor's previous grading history, so that we can judge how harsh or lenient a grader each instructor is. After making this determination, we adjust each grade given by that instructor to systematically correct for that instructor's bias.

After correction, the student body has an aggregate GPA of approximately 2.7, corresponding to an uninflated grade of B−. The corrected GPA values do a considerably better job of accurately ranking the students by ability, especially for students in the bottom eight deciles.

Assumptions

1. We wish to evaluate students purely based on their ability to perform well in courses.

2. Each student has a quality attribute that is not directly measurable but influences the ability to do well in courses. The ideal ranking of students is by highest quality attribute.

3. The instructor has an accurate perception of each student's performance in a course.

4. The cause of high grades is lenient grading practices by the average instructor at ABC College.

5. A more lenient instructor tends to grade all students higher, not just students of a certain ability level.

6. Scholarship selection is completed in the first half of a student's undergraduate career, to allow her to enjoy the scholarships while she is still in school.

7. Because the students are early in their careers at ABC College, they are still taking primarily general education courses, rather than courses in their major. Therefore, we assume that they select courses randomly, and thereby we model the breadth of course selection across disciplines.

8. Since the students know that their grades are going to be adjusted to filter out the harshness of their instructors' grading, they do not gravitate toward courses taught by lenient instructors.

9. Each student has a varying aptitude for each course. Presumably, a student has more aptitude for courses in her major. But since general course requirements tend to be broad and these are the courses we are examining, we assume that a student's aptitude for a course is random.

10. Each course has an inherent difficulty. In an easy course it is difficult to differentiate the high ability students from the rest, whereas tougher material produces a greater spread of performances.

11. Instructors know when a course is difficult. Presumably all students (even the top ones) will attain a lesser mastery of more difficult material, but the instructor will take this into account when assigning grades.

12. The college is on a semester system and each student takes four courses per semester.

13. A student's performance in a course is not influenced by which other students are taking the course. Neither is the student's grade, since we assume that instructors do not grade the students in a given course on a curve but rather on some absolute standard of performance.

14. An instructor's harshness in grading does not depend on the course and remains constant over a period of several years. Data on instructors' grading histories are available.

15. All instructors rate a student's performance the same, but they have different standards for what grade that performance should earn.

Practical Considerations

The concept of a single quality attribute that describes each student is not one that plays well politically and in the media. Not many people would advocate that a student's overall ability to do well in courses can be accurately characterized by a single real number. Therefore, our adjusted measure of student ability should be some sort of adjusted GPA, which will be easier for a general audience to accept and understand. This does not present a problem from the modeling point of view, as long as we know how quality rankings correspond to GPA values, and vice versa.

Ultimately, as we construct our model, we will run into a fundamental grading problem. The average grade at ABC College is an A−, which corresponds to a 3.67 GPA. Grade point averages that are this high result in very uninteresting grade distributions. The majority of the grades must be A+, A, or A−. In other words, if we look at the transcript of any above-average student at ABC, we will probably see a page full of A+, A, and A− grades. In this kind of environment, it will be extremely difficult to pick out the top few students, because the top half of the school is separated by only about 0.6 grade points. In contrast, the bottom half of the school is spread over the remaining 3.67 grade points, so it will be much easier to rank them by ability.

One radical solution to this dilemma is to require additional feedback on student performance from the instructors. We outline one possible system here, before we move on to a less radical approach. In addition to giving grades on the usual A to F scale, we could require an instructor to give each student a ranking between 1 and 10. At least one student in each course must receive a 1, and at least one student must receive a 10. This forces a spread in the instructor's rankings, so that even an easy-grading instructor (all A+ grades) must rank the better-performing students above the less able students. Next, the instructor is allowed to give a context to the scale. If the instructor has taught that course before, she would be asked to rate the current course in terms of previous ones. We ask the instructor to identify, on some absolute scale of ability, which interval corresponds to the 1 to 10 relative scale for the course. For example, if the instructor felt that her best student was about as competent as a student at the 90th percentile, then she would identify the right endpoint of the scale with the 90th percentile of absolute student ability. If she felt that her worst student was the poorest student to attend the college over an entire five-year span, then she would identify the left end of the relative scale with that point on the absolute scale. This two-stage evaluation system forces the students to be differentiated by performance puts the measures of performance into an absolute (rather than instructor-dependent) context.

What Characterizes a Good Evaluation Method?

As we attempt to rank the students at ABC College, we assume that the students have underlying quality scores that are reflected in their grades. We try to approximate the ranking induced by the hidden quality values. It may be inappropriate (for political reasons) to refer to our rankings as "estimated student qualities," so we instead calculate an adjusted GPA.

As we calculate adjusted GPA values, we keep in mind several goals:

- We wish to allocate correctly the available scholarships to the top 10% of the student body. To test whether or not we succeed, we must compare the ranking induced by our adjusted GPA values with the actual ranking of the students by intrinsic quality. Our first measure of the accuracy of our adjusted GPAs will just be the number of scholarships that we correctly allocated to deserving students.
- If the top-ranked student somehow fails to receive a scholarship, this is considerably more unjust than if a student who just barely deserves a scholarship misses out. Thus, we compute a second measure of accuracy by summing the severity of the mistakes made in awarding scholarships.
- It is important for all of the student rankings to be accurate, not just the top 10%, because they are used for much more than just scholarship determination. For instance, class rank is often cited in graduate school and job applications. Therefore, we consider a third measure of accuracy that gives a total error measure for our entire set of adjusted GPA rankings, rather than for just the top decile.

Modeling College Composition and Grade Assignment

According to Assumption 13, we do not need to consider the other students in a course when we determine a student's performance in the course and the grade the student receives; in other words, the composition of students in the course does not significantly affect the students' ability to learn, and none of the instructors grades on a curve. Thus, we model a student's grade as a function of

- her inherent quality,
- her aptitude for the specific course,
- the difficulty of the course, and
- the harshness of the instructor grading the course.

We treat each of these quantities as real-valued random variables and generate their values by computer.

We let q_i denote the inherent quality of student i. We will consider q_i to be distributed normally with mean 0 and standard deviation σ_q. This is reasonable, since we know that the normal distribution gives a good approximation for many characteristics of a large population.

We let $c_{i,j}$ represent the random course aptitude adjustment for student i when she takes course j. Again, it makes sense to let $c_{i,j}$ be normally distributed about 0, and we denote the standard deviation of this aptitude adjustment by σ_c. We let the net aptitude of student i in course j be $q_i + c_{i,j}$, which is normally distributed with mean 0 and standard deviation $\sqrt{\sigma_q^2 + \sigma_c^2}$. We choose our unit of measure so that $\sigma_q^2 + \sigma_c^2 = 1$. Furthermore, we estimate that a student's intrinsic quality influences her success at least five times as much as her aptitude adjustment for the particular course she is taking. Hence, we choose $\sigma_c < 0.2$.

Next, we consider how the difficulty of a particular course affects the grades that the instructor gives. We assume (see Assumption 10) that a difficult course spreads out the distribution of grades given; this means that poor students tend to do worse in difficult courses, but also that excellent students will do better, since they are being given an opportunity to excel. Conversely, in an easy course, the grades tend to bunch closer together, since the poor students are being given an opportunity to excel and the best students' performances are limited by the ease of the subject matter. Let d_j denote the difficulty of course j; then this interpretation leads us to consider a performance rating $N_{i,j}$ of student i in course j given by

$$N_{i,j} = (q_i + c_{i,j})d_j,$$

where d_j is a positive number, equal to 1 for a course of average difficulty, greater than 1 for a difficult course, and less than 1 for an easy course. Note that we are assuming the performance of a student in a course is random only in that the student's inherent ability is modified by a random aptitude adjustment factor. Once this factor is applied, the student's performance is determined, given the difficulty of the course.

Finally, we must take into account the grading philosophy of the instructor. Notice that a difficult course does not shift the performance distribution to the left, because the performance is measured relative to the instructor's expectation. We assume that the instructor is aware of the difficulty of the course and compensates accordingly in grading. This brings up a delicate distinction. An instructor's harshness does not reflect her expectation level but only her tendencies in grading. That is, we assume that the instructor's harshness does not pertain to her assessment of a student's performance but rather to what grade she thinks that performance deserves.

Let h_k denote the harshness of instructor k; then we should let the student's grade depend on $N_{i,j} - h_k$, since the harshness causes a systematic bias in all of the grades that the instructor gives. We let a harshness of 0 correspond to an average instructor at an institution without grade inflation. At Duke University and presumably at other institutions, 2.7 was the average GPA prior to the grade inflation that began to appear in the 1970s [Gose 1997]. Therefore, letting G denote the grading function that maps real numbers to discrete letter grades, we should center $G(0)$ on a grade of B$-$. Furthermore, among instructors who grade on a curve, an interval of one letter grade is often equated with one sample standard deviation in the course scores. So, we let one standard deviation for a course of average difficulty correspond to a whole letter grade in our model. Thus, the instructors in our model are grading on a virtual curve; that is, they grade on an absolute standard that simulates grading on a curve in a hypothetical course in which the full distribution of students is enrolled.

This analysis leads to a grading function

$$G : \mathbb{R} \to \{0, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13\}$$

defined by

$$G(x) = \begin{cases} 0, & \text{if } x \leq -\frac{11}{6}; \\ 3, & \text{if } -\frac{11}{6} < x \leq -\frac{9}{6}; \\ 4, & \text{if } -\frac{9}{6} < x \leq -\frac{7}{6}; \\ \vdots & \\ 12, & \text{if } \frac{7}{6} < x \leq \frac{9}{6}; \\ 13, & \text{if } x > \frac{9}{6}, \end{cases}$$

where the values $0, \ldots, 13$ represent the letter grades F, D, D$+$, C$-$, C, C$+$, B$-$, B, B$+$, A$-$, A, A$+$. To convert the numeric value to grade points, we divide by 3; thus, an A$-$ average means a GPA of 3.67. If student i takes course j taught by instructor k, she will receive a grade of

$$G(N_{i,j} - h_k) = G\big((q_i + c_{i,j})d_j - h_k\big).$$

For convenience, we define $l(g)$ and $r(g)$ to be the left-hand and right-hand endpoints of the interval on which $G = g$. For instance, $r(13) = \infty$ and $l(12) = \frac{7}{6}$.

A simple calculation reveals that in a course where $d = 1$ and $h = 0$, the expected grade is 2.63 (see **Table 2** and **Figure 1**). We intend harshness 0 to represent a reasonable level of strictness in grading. It centers the grades at B$-$, which is the exact middle of all passing grades, and yields a GPA in line with the "reasonable" historical number of 2.7 at Duke University.

We can visualize the grading method by graphing a $\mathrm{normal}(0, 1)$ density function, which represents N (in the case where difficulty is 1), with the x-axis partitioned into intervals representing grades according to the grading function G (see **Figure 2**). A difficult course spreads out the distribution, resulting in more Fs (because the poor students cannot keep up) and more As (because the top students have an opportunity to shine). A positive (negative) harshness effectively shifts the grade intervals to the right (left).

Since we have no data on the students, courses, instructors, or grades at ABC College, we generated a random set of students, courses, and instructors. Since we don't know the exact composition of the college, we generated various scenarios.

- We assigned each instructor to teach five courses per year, which is a typical teaching load at many colleges.

Table 1.

Symbol table.

$c_{i,j}$	course aptitude adjustment for student i taking course j
d	estimate of course difficulty
d_j	difficulty of course j
G	grading function, from performance rating to letter grade
$\overline{g}$	average grade given by instructor, from historical data
g_{adj}	adjusted grade that an instructor of harshness zero would give
h_k	harshness of instructor k
I	interval in which student's performance value is estimated to lie
$l(g), r(g)$	endpoints of performance rating interval corresponding to letter grade g
$N_{i,j}$	performance rating of student i in course j
N_{est}	estimate of student performance value
$N_{i,\text{est}}$	estimate of student i's performance value
Φ	standard normal cumulative distribution function
q_i	inherent quality of student i
$q_{i,\text{est}}$	estimate of inherent quality of student i
σ_q	SD of inherent quality of student i
σ_i	SD of course aptitude adjustment of student i taking course j
$\sigma(d,h)$	SD of grades given by a professor of harshness h in a course of difficulty d
t_0	left endpoint for performance rating interval corresponding to grade of A+
t_1	right endpoint for performance rating interval corresponding to grade of F

Table 2.

Expected grade on a 4-point scale as a function of course difficulty d and instructor harshness h.

	d										
h	0.5	0.6	0.7	0.8	0.9	1.0	1.1	1.2	1.3	1.4	1.5
−3.0	4.33	4.33	4.32	4.31	4.30	4.29	4.27	4.25	4.23	4.20	4.18
−2.8	4.33	4.32	4.31	4.30	4.28	4.27	4.24	4.22	4.19	4.16	4.13
−2.6	4.32	4.31	4.30	4.28	4.26	4.24	4.21	4.18	4.15	4.12	4.08
−2.4	4.31	4.30	4.28	4.25	4.22	4.19	4.16	4.13	4.10	4.06	4.03
−2.2	4.29	4.27	4.24	4.21	4.18	4.14	4.11	4.07	4.03	4.00	3.96
−2.0	4.26	4.22	4.19	4.15	4.11	4.08	4.04	4.00	3.96	3.92	3.88
−1.8	4.19	4.15	4.11	4.07	4.03	3.99	3.95	3.91	3.87	3.83	3.79
−1.6	4.10	4.06	4.02	3.98	3.94	3.90	3.86	3.82	3.78	3.73	3.69
−1.4	3.97	3.94	3.90	3.86	3.82	3.78	3.74	3.70	3.66	3.62	3.58
−1.2	3.82	3.79	3.76	3.72	3.69	3.65	3.62	3.58	3.54	3.50	3.46
−1.0	3.64	3.62	3.60	3.57	3.54	3.51	3.48	3.44	3.41	3.37	3.33
−0.8	3.45	3.44	3.43	3.41	3.38	3.35	3.32	3.29	3.26	3.23	3.19
−0.6	3.26	3.25	3.24	3.23	3.21	3.19	3.16	3.13	3.10	3.07	3.04
−0.4	3.06	3.06	3.05	3.04	3.03	3.01	2.99	2.96	2.94	2.91	2.88
−0.2	2.86	2.86	2.86	2.85	2.84	2.82	2.80	2.78	2.76	2.74	2.72
0.0	2.66	2.66	2.66	2.65	2.64	2.63	2.61	2.60	2.58	2.57	2.55
0.2	2.46	2.46	2.46	2.45	2.44	2.43	2.42	2.41	2.40	2.39	2.38
0.4	2.26	2.26	2.25	2.24	2.23	2.23	2.22	2.21	2.21	2.20	2.20
0.6	2.06	2.05	2.04	2.03	2.03	2.02	2.02	2.02	2.02	2.02	2.02
0.8	1.85	1.84	1.83	1.82	1.81	1.81	1.82	1.82	1.83	1.84	1.85
1.0	1.63	1.62	1.61	1.60	1.60	1.61	1.62	1.63	1.64	1.66	1.67

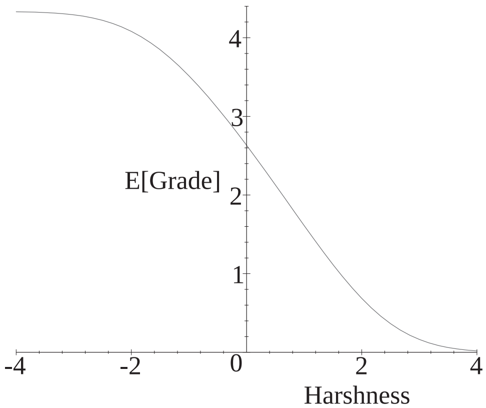


Figure 1. Expected grade on a 4-point scale as a function of instructor harshness h, given that course difficulty $d = 1$.

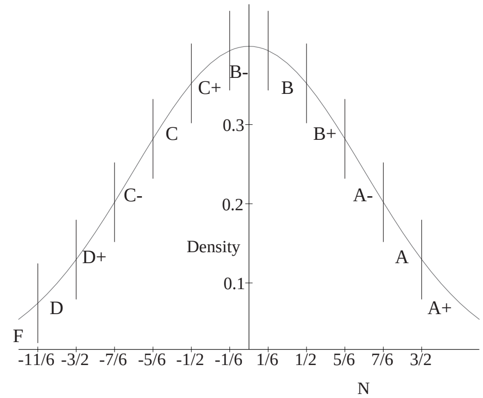


Figure 2. The probability density of the performance variable N. The vertical bars represent the grade ranges for an instructor of zero harshness. For $h > 0$, the ranges shift to the right by h, making it harder to earn a high grade.

- We computed a quality variable q for each student by sampling randomly from a $\mathrm{normal}(0, \sigma_q)$ distribution.
- We assigned each student to eight courses per year (also a standard load for many colleges), for either one or two years, using a uniform probability of selecting each course.
- We generated course aptitude adjustments c for each course that a student enrolled in by sampling from a $\mathrm{normal}(0, \sigma_c)$ distribution.
- We assigned difficulties to each course by sampling from a symmetric beta distribution centered at 1. A typical choice would be $\mathrm{beta}(3, 3)$ on $[0.7, 1.3]$ (see **Figure 3**).
- We assigned harshnesses to instructors by sampling from an asymmetric beta distribution skewed and translated towards leniency (to represent the tendency to inflate grades at the college). A typical choice would be $\mathrm{beta}(2, 3)$ on $[-2, 0]$ (see **Figure 4**). One can use **Table 2** to guide the choice of distribution according to the average GPA that we desire.

We did not restrict ourselves to considering the case where the average GPA at the college is 3.67. In early 1997, Duke University considered revising its calculation of GPAs to take into account the difficulty of the course in which a grade was received, the quality of the other students in the course, and the historical grading tendencies of the instructor. The reason for this move was alarm that the average GPA at the University had risen from 2.7 in 1969 to 3.3 by fall 1996 [Gose 1997]. If an average GPA of 3.3 is considered evidence of rampant grade inflation, then 3.3 is a more likely estimate of the average GPA at ABC College than 3.67 is.

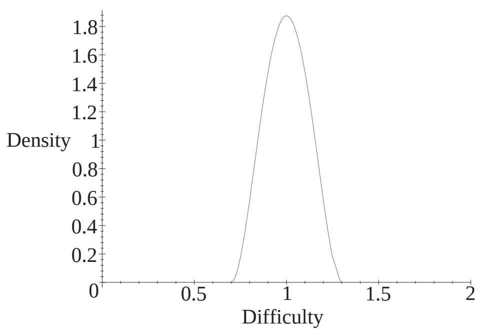


Figure 3. Typical course difficulty distribution: $\mathrm{beta}(3,3)$ on $[0.7,1.3]$.

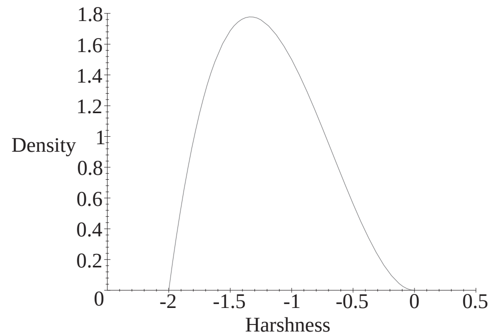


Figure 4. Typical instructor harshness distribution: $\mathrm{beta}(2,3)$ on $[-2,0]$.

Some word on our rationale for choosing distributions is in order here. The normal distribution is a standard choice for representing abilities in a population. Our model of how course difficulty affects student performance and grades loses validity for d outside the range of approximately $[0.5, 2]$, and of course a negative difficulty makes no sense at all. Thus, the normal distribution is not an appropriate choice. We chose a beta distribution because it takes values on a finite interval. Similarly, any harshness value outside the range $[-3, 2]$ is patently ridiculous (see **Figure 1**). In fact, a harshness value of -2 is fairly ridiculous; but to obtain a school-wide average GPA of 3.67, we have to allow that some instructors are that lenient. In any event, it behooves us to choose a distribution over a finite interval. We also desire an asymmetric distribution, to represent the tendency at the college toward lenient grading. Thus, a $\mathrm{beta}(a, b)$ distribution with $a < b$ is appealing for our purposes.

Our model indicates that the ABC College administrators' concerns about being unable to distinguish among the top students are justified. Indeed, when we generate an incoming freshman class of 500 students and make the instructors lenient enough to yield a 3.64 average GPA, 60 of these students *still have a straight A+ average after two years!* In this environment it is clearly necessary to search for a better evaluation method than simple GPA.

The Modified GPA algorithm

Our algorithm for establishing a class rank involves a number of distinct stages. We first attempt to gain additional information from the instructor's historical grade awards and use this information to refine our knowledge of the courses and instructors that the students are taking currently. Using an estimate of the instructor's harshness based on the grades that he has given historically, we correct the grades awarded in a particular course by estimating the mean value by which any leniency or harshness changed a student's letter grade. Incorporating this correction factor allows us to provide an adjusted

GPA measure that represents more fully the actual performances and, hence, the quality of the students.

We assume that the instructors have been teaching at the college for at least two years and that we have access to the grades that they assigned during those years. We simulate these data just as we generated the data for the students, as described in section **Modeling College Composition and Grade Assignment**. New students are generated randomly as in the original student data. Given the actual harshness of the instructors and the difficulties of the courses taught, numerical performances and grades for all of each instructor's courses are generated as above.

From these historical data, we compute the average grade $\overline{g}$ granted by each instructor. One can calculate the expected grade granted in a course as a function of difficulty d and harshness h (see **Table 2** and **Figure 1**). For a given value of d, this function decreases monotonically with h, so we can calculate the inverse function. Assuming that $d = 1$ we estimate the instructor's harshness h_{est} by evaluating this inverse function at $\overline{g}$.

Notice that we never even try to estimate the difficulty or take the actual grade distribution into account. Despite this crude method of estimating harshness, we achieve surprisingly good results. Using courses of 40 students each, the harshness that we estimate is usually within about 0.05 of the actual harshness, though it is not too uncommon to err by as much as 0.12. The error tends to decrease the closer the actual harshness is to zero.

We can now adjust the grades of the students in each of an instructor's courses based on our harshness estimate h_{est} for that instructor. For simplicity we assume $d = 1$ for the course. The fact that a student receives a grade g in a course with a instructor of harshness h means that the student's performance value N lies in the interval

$$(l(g) + h,\ r(g) + h]\ .$$

Thus we estimate that N lies in the interval

$$I = (l(g) + h_{\text{est}}, r(g) + h_{\text{est}}]\,.$$

We estimate N to be the expected value of the distribution of N given that N lies in I. For grades other than A+ and F, this interval has width $\frac{1}{3}$. Assuming $d = 1$, the a priori distribution of N is standard normal. Given that it lies in I, the probability density function is just the indicator function for I times the standard normal density times a constant factor. Since the density function for the standard normal distribution is almost linear over any interval of width $\frac{1}{3}$, we estimate N as

$$N_{est} = h_{\text{est}} + \frac{l(g) + r(g)}{2}.$$

If the grade is A+ or F, we can calculate the expected value analytically, with

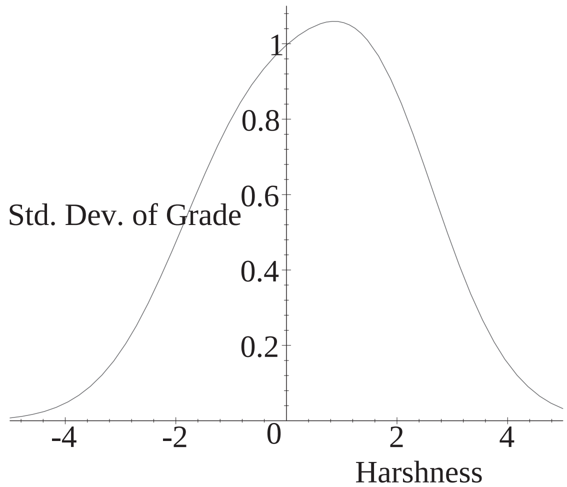


Figure 5. Standard deviation of grade distribution as a function of h, assuming difficulty $d = 1$.

the following results:

$$E[N|N > t_0] = \frac{e^{-t_0^2/2}}{\sqrt{2\pi}(1 - \Phi(t_0))} \tag{1}$$

$$E[N|N \le t_1] = -\frac{e^{-t_1^2/2}}{\sqrt{2\pi}\Phi(t_1)} \tag{2}$$

where Φ is the standard normal cumulative distribution function and the relevant t values are $t_0 = l(A+)$ and $t_1 = r(F)$. So when the student's grade is A+ or F, we set N_{est} to **(1)** or to **(2)**, respectively.

Now that we have a value for N_{est}, we assign the student an adjusted grade for the course. The adjusted grade g_{adj} is the grade that an instructor of zero harshness would have given, except that we assign a real number grade instead of an integer. Specifically,

$$g_{\text{adj}} = 3N_{\text{est}} + 8.$$

To avoid a discontinuity at $N_{\text{est}} = r(F)$, we treated an F as a grade of 2 for this purpose.

Since the grades given by a lenient grader exhibit a smaller spread and hence do not differentiate the students as well as those given by a strict grader, we grant them less import when calculating a student's adjusted GPA. Specifically, the student's GPA is a sum of the grades received weighted by $\sigma(1, h_{\text{est}})$, where h_{est} is the estimated harshness of the instructor who assigned the grade and $\sigma(d, h)$ is the standard deviation of grades given by a instructor of harshness h in a course of difficulty d (see **Figure 5**).

If we wished to refine this method, we might use the spread of grades in a course to estimate its difficulty both when examining the instructors' grading histories and when adjusting grades at the end. However, even without this enhancement, the basic method that we have just outlined performs well, as we demonstrate in **Results of the Model**. One has to be very clever to estimate the difficulty of a course in a way that is numerically stable. The most obvious method is to note that for student i, we have $N_i = q_i d$, then use $N_{i,\mathrm{est}}$ and some estimate $q_{i,\mathrm{est}}$ of q_i that we pull from some other source to estimate

$$d \approx \frac{N_{i,est}}{q_{i,est}}.$$

The difficulty is that q_i is likely to be near zero, so the error $|q_i - q_{i,\mathrm{est}}|$ is magnified when we divide.

Results of the Model

We generated a number of scenarios to elucidate the features both of our simulated data and of the modified GPA algorithm. We ran these simulations on a test student population of 500 students, each of whom took 16 courses, with average course size 40.

We use a number of these scenarios to demonstrate the results of our simulations. Plots of actual quality ranks vs. rank by GPAs or adjusted GPAs demonstrate the effectiveness of each ranking method over all tiers of students. For a perfect ranking of students, this plot would lie along the line $y = x$.

We define three error metrics to aid in the comparison between the ranking generated by our revised GPA method and the raw GPA ranking.

- We define a *misassigned scholarship candidate* to be a student who either received a scholarship but was not in the top 10% in quality, or who was in the top 10% of quality but did not receive a scholarship. A simple count of the number of misassigned scholarship candidates measures the method's effectiveness at identifying the highest caliber of student. We refer to this quantity as the MS (Missed Scholarship) metric for a given estimated ranking.

- For each student who is ranked incorrectly, the rank errs by some number of places. Summing these rank errors over all students gives us a measure of how our ranking compares to the actual quality ranking across the entire spectrum of students. We refer to this as the SE (Scaled Error) metric.

- Finally, to determine the injustice with which scholarships are assigned, we sum over all misassigned scholarship candidates the distance between their quality ranking and the scholarship cutoff rank. We refer to this measure of error as the SI (Scholarship Injustice) metric.

The first scenario has difficulty scaled to be between 0.7 and 1.3, while the harshness distribution is relatively lenient, with values ranging between -2.1 and -0.1. The variation due to course material is set to be 0. This yields, as one might expect, a student population with rampant grade inflation. Overall GPA is 3.64, with 60 students receiving perfect A+ averages. **Figure 6** plots GPA rank against actual quality rank, where we observe significant discrepancies between the estimated and actual rankings. At this level of grade inflation, the top tiers of students are almost entirely indistinguishable by GPA. Attempting to correct for harshness by using the corrected GPA does not significantly improve the results at the high ranks. It does, however, improve the SE number from 9,124 to 5,352, representing a superior evaluation of the middle and lower tiers of students (see **Figure 7**).

We now alter the parameters in our model to fit what we feel is a far more realistic situation. Harshness is set to vary between -1.569 and $.431$, yielding a scenario that has an average GPA of 3.34. Now only 38 students have perfect A$+$ averages. The effect of ranking based on the adjusted GPA is definitely apparent, in **Figures 8–9**. The discrepancies are clearly less for middle-ranked and low-ranked students. The SE measure improves from 6,916 using simple GPAs to 4,842 using adjusted GPAs.

Note the loss of ranking accuracy at high quality levels. The two methods of ranking perform nearly identically, with raw GPA giving an MS of 6 while the adjusted GPA rank gives an MS of 7.

Table 3 summarizes results for a sampling of the scenarios, all of which use the usual range of difficulty from 0.7 to 1.3.

As our simulations show, the modified GPA ranking fares well against the raw GPA method, with SE numbers significantly lower in each trial. This means that the students in the lower deciles are ranked much more accurately in each case. This suggests a certain robustness and indicates that judgments based on the modified GPA rank will be more fair.

As in all other trials performed, both the raw and adjusted GPA ranking methods performed poorly at the high end of the ability curve, according to all three measures.

Strengths and Weaknesses

One weakness of our model is that it does not allow for a completely analytic solution to the scholarship selection problem. Computer simulation is the only means we have to test and evaluate our methods against the simple-minded raw GPA ranking method.

Potentially the greatest weakness in our model and techniques is the lack of a good ranking of the top two deciles. Whether or not a robust method exists is, we believe, debatable. Nothing we have seen indicates that the information required to form a confident ability ranking is even contained in the GPA information we have. It is likely that complete rank-ordering cannot be achieved given the information our model provides. We do not know this to be true, but it is certainly consistent with the results we have witnessed.

Another weakness is that our model cannot take into account the effect of

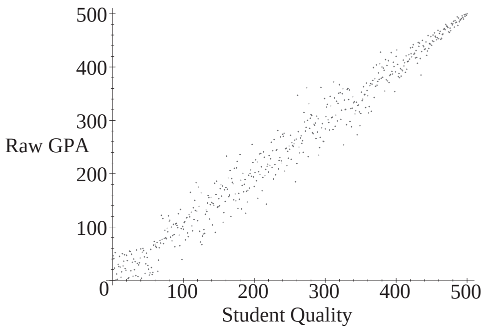


Figure 6. Wild grade inflation resulting in an average GPA of 3.64. The raw GPA estimate makes significant mistakes in the entire range but is especially inaccurate in the top two deciles.

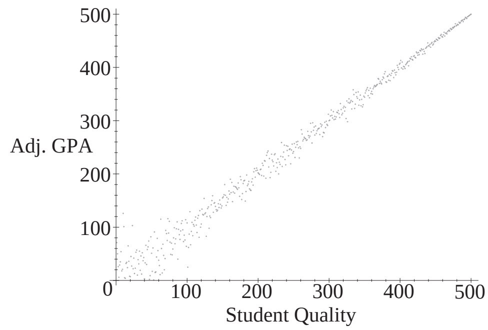


Figure 7. The same scenario as **Figure 6** with rank determined by GPAs modified for instructor harshness.

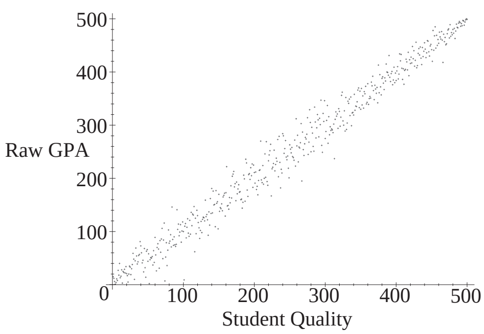


Figure 8. A more reasonable scenario. The raw GPA rank maintains some level of inaccuracy throughout the spectrum of student ability.

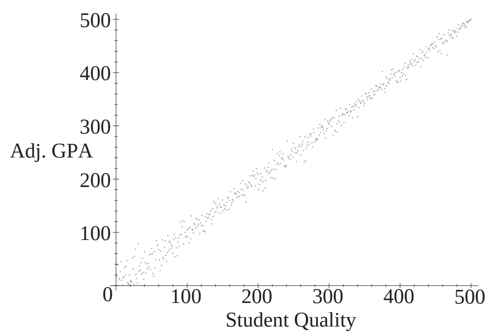


Figure 9. Same scenario as **Figure 8** but with ranks determined by the modified GPAs.

Table 3.

The relevant information for several simulations. Note how the modified GPA ranking produces smaller SE numbers in all cases, representing greater overall accuracy.

Trial	Harsh low	Harsh high	Raw GPA	Raw MS	Adj. MS	Raw SE	Adj. SE	Raw SI	Adj. SI
1	−2.1	−0.1	3.64	9	16	9124	5352	139	118
2	−1.55	0.35	3.22	8	10	7178	3122	247	596
3	−1.569	0.331	3.34	6	7	6916	4842	215	197
4	−1.9	0.1	3.48	8	10	8896	5424	101	226
5	−1.49	0.51	3.19	8	5	7454	4410	153	81

curved grading systems and the possibility of student grades being altered by the performances of the fellow students in the course. Similarly, other interactions between the entities in our model, such as the formation of study groups, can affect the performance (as distinguished from grade) of a student in a course in a way that is dependent upon the other members of the course. Our model also includes parameters, namely, the course difficulties, that are difficult to estimate accurately, and thus remain a complete unknown throughout our attempts to rank based on ability.

In spite of these features, our model has a number of compelling features. By changing just a few parameters, one can generate an entirely new scenario that has a plausible distribution of grades and GPAs. Furthermore, it takes into account the three functional parts of any educational experience: the students, the instructors, and the courses. Arguably, no model could be complete without accounting for variations of each of the three parts.

Despite all of our problems in classifying the scholarship winners, the adjusted GPA method we use is almost uncannily good at identifying the lower deciles, which in a real context is important to the students and the school.

From a practical standpoint, our model and methods are fairly simple to implement. The number and size of the calculations performed is linear in the size of the student body and could be executed with modest computer resources at even a large institution.

To sum up, it would behoove ABC College to use our ranking system, since it more accurately identifies the bottom eight deciles of student ability. However, if the administration seeks to accurately rank the top tier of students, it must realize that a bloated aggregate GPA from excessively lenient grading can quickly lead to a situation where no amount of calculations and statistics can recover the desired information about the intrinsic quality of the students.

References

Gose, Ben. 1997. Duke rejects a controversial plan to revise the calculation of grade-point averages. *Chronicle of Higher Education* 43 (21 March 1997): A53.

1999: The Asteroid Impact Problem

For some time, the National Aeronautics and Space Administration (NASA) has been considering the consequences of a large asteroid impact on the earth.

As part of this effort, your team has been asked to consider the effects of such an impact were the asteroid to land in Antarctica. There are concerns that an impact there could have considerably different consequences than one striking elsewhere on the planet.

You are to assume that an asteroid is on the order of 1,000 m in diameter, and that it strikes the Antarctic continent directly at the South Pole.

Your team has been asked to provide an assessment of the impact of such an asteroid. In particular, NASA would like an estimate of the amount and location of likely human casualties from this impact, an estimate of the damage done to the food production regions in the oceans of the southern hemisphere, and an estimate of possible coastal flooding caused by large-scale melting of the Antarctic polar ice sheet.

Comments

The Outstanding papers were by teams from Harvey Mudd College (two papers), Pacific Lutheran University, the University of California–Berkeley, and the University of Puget Sound. Their papers, together with a judge's commentary, were published in *The UMAP Journal* 20 (3) (1999): 211–271.

Problem Origin

The problem was contributed by Jack Robertson (then at Georgia College and State University, currently at Georgia Military College).

Judge's Comments

Pat Driscoll (U.S. Military Academy) remarked that there seemed to be only a handful of approaches.

Using the Internet as a source of information proved to be a two-edged sword. Sites such as those at Sandia Laboratories or the Jet Propulsion Laboratory provided interesting and in some cases accurate and relevant information dealing with asteroid impacts with Earth. Unfortunately, teams extracting information from these sites without first thinking about and discussing the problem soon found themselves under the spell of the siren, lulled into a mathematical approach that they could not bring to successful closure in the time allotted for the competition. Moreover, to judge from their lack of direct supporting documentation and reasoning, in the end they found themselves unprepared to explain clearly and sufficiently the underlying assumptions and reasoning of the mathematics presented at the sites. As in most modeling efforts, this provided an all-too-fatal flaw.

Each Outstanding team chose one or two techniques (e.g., partial differential equations, kinetic energy modeling, etc.) to develop within the context of a complete modeling effort. They provided conclusive evidence that they had dedicated a substantial amount of time thinking about the problem prior to the quest for supporting information. This choice enabled them to weigh the cost and benefit of identifying exact modeling parameters versus making reasonable assumptions and working with approximate, in-range values. The many facts directly associated with the problem—such as the geological composition of both the asteroid and Antarctica, the typical source of Earth-bound asteroids, the angle of incidence upon impact, the human population distribution, and atmospheric currents and circulation—mandated such a strategy. Papers failing to provide evidence of having considered important problem characteristics, whether implicit or explicit, were eliminated from further consideration. As a minimum, it would have been better to identify and explain the impact of a particular feature (e.g., upper atmospheric wind currents) and then to choose explicitly not to include this factor for reasons of mathematical tractability.

Modeling assumptions fell into two broad categories:

- physical assumptions requiring justification with discussion, and
- numerical parameter values from citations noted.

The plausibility and applicability of either type directly depended on how well teams linked a particular assumption to the problem as stated in the MCM, rather than to some problem stated in the reference source document. Regardless of a paper's calculations, a 10-m instantaneous rise in all of the Earth's oceans is a bit too far-fetched of a result for the problem presented, even for the most devout of science fiction followers to accept.

The exceptional papers all conveyed a clear link to verifiably credible information sources. Lesser-quality papers showed a reliance on Internet sites for supporting information and failed to include necessary explanations of why certain parameter values were valid and what assumptions their methods were based on. Although the temptation to "cut-and-paste" directly from Internet sources is recognizably strong, doing so most often resulted in a paper that was predominantly statements of unsupported "facts" rather than one showing that the team had a clear understanding of the model. Additionally, dedicating an inordinate amount of time to display the derivation of known relationships (e.g., Kepler's law of gravitational attraction) added little value to a paper.

The Sky is Falling!

Daniel Forrest
Garrett Aufdemberg
Murray Johnson
University of Puget Sound
Tacoma, WA 98416

Advisor: Perry Fizzano

Assumptions

1. The diameter (D) of the asteroid at impact is 1,000 m. Heat and stress while traveling through the Earth's atmosphere would cause some portion to vaporize or burn before impact. However, for an object this large traveling at speeds typical of cosmic objects impacting the earth, one can ignore the deceleration and ablation (loss of mass from the surface of an object due to frictional forces) due to the atmosphere [Steel 1995, 178].

2. The asteroid strikes the earth at the geographic South Pole.

3. The asteroid is spherical.

4. The asteroid is homogeneous with uniform density $\rho = 2.5$ g/cm^3; uniform density allows for simple estimates of the mass. The value of ρ is typical of C-type (carbonaceous) asteroids, which make up the majority of the asteroids in the solar system and therefore are the most likely type to strike earth, and also within the typical range of densities of S-type (stony) asteroids, which make up a majority of the asteroids with orbits that cross the Earth's orbit [Morrison and Owen 1996, 103–132].

Preliminary Calculations

Mass of the Asteroid

The mass of the asteroid (M_a) is its density (ρ) multiplied by its volume (V). For a spherical asteroid, the mass is given by

$$M_a = V\rho = \frac{4}{3}\pi\left(\frac{D}{2}\right)^3\rho.$$

For our asteroid, $D = 1{,}000$ m and $\rho = 2.5$ g / cm^3, thus

$$M_a = 1.3 \times 10^{12} \text{ kg}.$$

Upper and Lower Bounds on Impact Speed

A planet's escape velocity (v_{esc}) is the minimum speed that an object must have to escape the planet. It is calculated by determining the change in potential energy caused by moving an object from the planet's surface to "infinity." To escape the planet, the object's initial kinetic energy must be greater than or equal to the change in potential energy. By symmetry, the escape velocity is also the minimum velocity that an object from beyond the planet can have when it reaches the planet's surface. Thus, the Earth's escape velocity, $v_{\text{esc}} = 11.2$ km / s, is a lower bound on the asteroid's impact speed (v_{imp}).

There is also an upper bound on the impact velocity, "a combination of escape velocity, heliocentric orbital velocity, and the velocity of an object just barely bound to the sun at the planet's orbital position." For Earth, this maximum is 72.8 km / s [Melosh 1989, 205]. Thus, the impact velocity is bounded by

$$11.2 \text{ km / s} \le v_{\text{imp}} \le 72.8 \text{ km / s}. \quad \textbf{(1)}$$

Energy Released on Impact

The energy of the collision (E_{imp}), drawn from the kinetic energy of the asteroid, is

$$E_{\text{imp}} = \frac{1}{2}M_a v_{\text{imp}}^2.$$

The impact velocity is bounded and the asteroid's mass is fixed. Applying **(1)**, we have

$$8.2 \times 10^{19} \text{ J} \le E_{\text{imp}} \le 3.4 \times 10^{21} \text{ J}. \quad \textbf{(2)}$$

Effects of Impact

Crater Size

The crater from the impact would be roughly parabolic in shape, with a diameter of approximately 10 km and a depth of approximately 1 km [Koeberl and Sharpton 1998]. The pressure is so great in impacts of this sort that the crater forms partially from the vaporization of the target material. At the South Pole, the asteroid would be impacting in ice about 2,600 m thick. It takes considerably lower energies to vaporize ice than rock or soil, therefore we expect that the impact crater would be larger than similar impact craters in other locations.

Melting and Vaporization of Antarctic Polar Ice Cap

Could an asteroid impact at the South Pole melt the Antarctic polar ice cap and drastically changing global sea levels? The ice cap covers 1.32×10^{13} m^2 with average thickness $2{,}440$ m [Ronne 1997]. Thus, there is 3.2×10^{16} m^3 of ice, with mass 2.9×10^{19} kg.

At most, the asteroid impact could create 3.4×10^{21} J. If all the energy were to melt ice, how much ice could be melted?

Assuming that the ice is at $0°$ C, it would take 3.33×10^5 J/kg to melt 1 kg of ice [Wilson and Buffa 1997]. So, at most

$$\frac{3.4 \times 10^{21}}{3.3 \times 10^5} \approx 1 \times 10^{16} \text{ kg}$$

of ice could be melted. This translates to 1×10^4 km^3 of liquid water. The area of the world's oceans is approximately 3.61×10^6 km^2; so if the melted water were evenly distributed across the world's oceans, sea level would rise less than 3 cm. This is not enough to endanger human lives or displace human settlements.

This estimate is an upper bound, since some energy goes into destroying the asteroid on impact; vaporizing part of the asteroid; vaporizing ice; excavating the crater; creating sound, shock, and seismic waves; and heating the air around the impact site. The impact would probably vaporize much of the ice from the impact crater. Assuming that the volume of ice vaporized is equal to the volume of ice in the largest cone that fits in the roughly parabolic crater, the impact would vaporize 2.6×10^{10} m^3 of ice, or 2.4×10^{13} kg of ice. The energy required to melt a kilogram of ice, heat the kilogram of resulting water to $100°$ C, and vaporize the water is 3×10^6 J. Vaporizing so much ice would require 7.2×10^{19} J. This value is within the bounds on the impact energy in **(2)**.

Earthquakes and the Risk of Tsunami

We can estimate the magnitude (Q) of the seismic disturbance (as measured on the Richter scale) from the formula [Melosh 1989, 67]

$$Q = 0.67 \log_{10}(E_{\text{imp}}) - 4.87. \tag{3}$$

The seismic disturbance due to a cosmic impact is not the same as from normal seismic activity. The effect of impact-generated seismic waves is estimated to be an earthquake of one magnitude less than the approximate magnitude generated by impact [Melosh 1989, 67].

For our asteroid, equation **(3)** (using the energy range from **(2)**) tells us that the impact would generate a seismic disturbance ranging in magnitude from 8.5 to 9.6 on the Richter scale (**Figure 1**). Even if the effects are discounted by one magnitude, such an earthquake would cause many human casualties if located in a more-populated part of the world than the South Pole. However, human casualties are negligible because the continent is mostly uninhabited and because Antarctica is large enough that any damage would be limited to Antarctica.

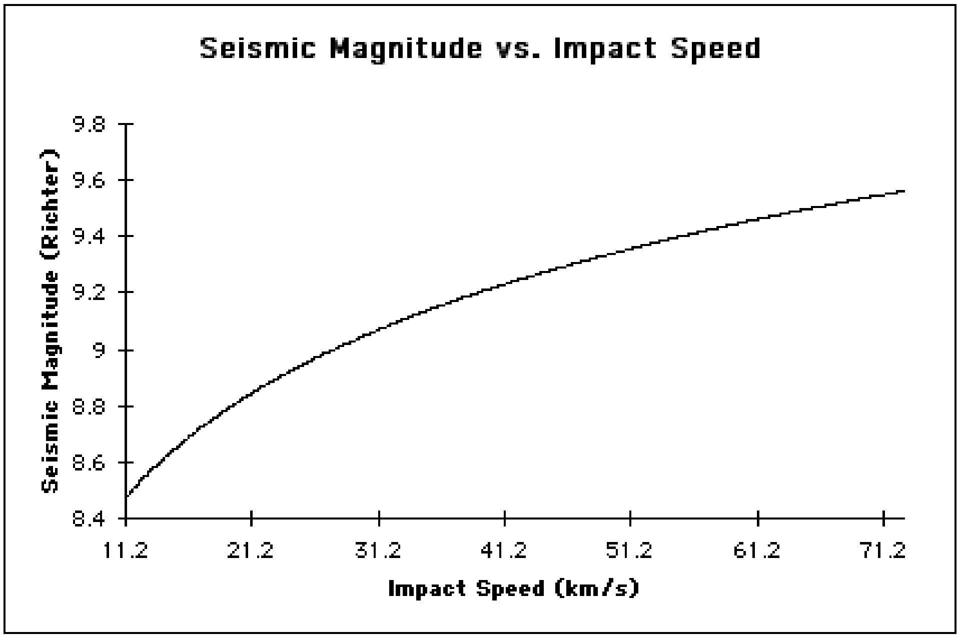


Figure 1. Seismic magnitude vs. impact speed.

Because the impact is at least 500 km from the closest shoreline and 1,500 km from most of the shoreline, the risk of a catastrophic tsunami being generated is negligible. A large percentage of the coast of Antarctica is lined with sheer walls of ice (on the order of 30 m in height). There is indeed a very real danger that the seismic disturbance could cause large fragments to break off, fall into the water, and cause tsunamis. Landslide-generated tsunamis can be large; the 1936 tsunami in Lituya Bay, Alaska, reached a height of 150 m [Hamilton 1998a]. However, they dissipate quickly and are unable to cross the great, transoceanic distances associated with earthquake-generated tsunamis. The greatest risk would be to coastal areas on the southern tip of South America.

Atmospheric Effects

Upon impact, the asteroid would disintegrate. Approximately 10% of the mass, 3.1×10^{11} kg, would be vaporized into submicron particles that would rise to the stratosphere (an altitude of 16 to 48 km) and would remain there for months [Steel 1995, 67]. If dust made up of 1-micron particles were spread evenly in a 1-micron-thick spherical layer at height H above the surface of the earth, it would cover approximately 10% of the surface area of the imaginary sphere and would block 10% of incoming solar radiation. On a very cloudy day, the intensity of light reaching the surface of the earth is roughly 10% of the intensity of light on a clear day [Steel 1995, 66]. A 10% drop in intensity would allow 9 times the intensity of light to reach earth as on a very cloudy day; but over a period of months, such a drop would be significant enough to cause global temperature change.

The ice vaporized on impact would rise into the atmosphere and form clouds. The water vapor in these clouds would eventually fall to earth as rain, increasing the amount of liquid water on the Earth by 8.1×10^{10} m^3. If it all ended up in the world's oceans, the global sea level would rise about 2 cm.

Conclusions

Fear that the ice cap would melt and cause global flooding is unfounded.

Because the asteroid would impact at the South Pole, the dust levels are far less than if the same asteroid impacted in soil and / or rock. Still, enough dust is lifted into the stratosphere to block up to 10% of the sunlight—enough to impact global temperature but far from the threshold where photosynthesis becomes impossible. Reduced light levels and temperature would affect agricultural production, but the impact on the world's food supply would be small; food surpluses in industrialized countries should be able to make up for agricultural losses in other nations.

The ice vaporized from the crater would form clouds and eventually fall to earth in liquid form. But the volume of the water is not large enough to cause large-scale coastal flooding, unless it all falls in a limited area in a limited amount of time. The dust that is larger than a micron and does not reach the stratosphere could still have detrimental effects, such as acid rain. But our model has no way of estimating the amount, location, or effects of possible acid rain.

Because the asteroid hits in Antarctica, the death toll directly due to impact is limited to the few hundred researchers stationed there. These casualties could be eliminated by evacuation if there is enough advance warning.

Strengths and Weaknesses

Our model is successful in that we have quantitative estimates of many of the effects associated with the impact, such as the range of possible impact velocities, the range of possible impact energies, the size of the impact crater, the effect of dust raised by impact in the atmosphere, and the magnitude of seismic disturbance generated by impact. Our model is simple enough that all calculations were performed without resorting to a computer.

The simplicity of our model also brings about some weaknesses. We have no accurate method to estimate how the total impact energy is distributed. We are also unable to determine long-term environmental consequences. Because of the unpredictable nature of atmospheric dynamics, we are unable to develop a model that would show specific locations and amounts of crops affected by dust raised from the impact. Our model predicts no direct loss of human life, but we are unable to take into account human life lost due to effects on food production.

Our model, while not sophisticated, offers intuitive results. Our estimates of crater size, impact energy, and magnitude of seismic disturbance correlate nicely to other models' predictions, such as those of Hamilton [1998b].

References

Fishbane, Paul M., Stephen Gasiorowicz, and Stephen T. Thornton. 1993. *Physics for Scientists and Engineers*. Englewood Cliffs, NJ: Prentice Hall.

Hamilton, Douglas P. 1998a. Notable Tidal Waves of this Century. `http://janus.astro.umd.edu/astro/Wave.htm` .

________. 1998b. Solar System Collisions. `http://janus.astro.umd.edu/astro/impact.html` .

Koeberl, Christian, and Virgil L. Sharpton. 1998. Terrestrial Impact Craters. `http://cass.jsc.nasa.gov/publications/slidesets/impacts.html` .

Melosh, H.J. 1989. *Impact Cratering: A Geological Process*. New York: Oxford University Press.

Morrison, David, and Tobias Owen. 1996. *The Planetary System*. Reading, MA: Addison-Wesley.

Ronne, Finn. 1997. Antarctica. *Encyclopedia Americana*.

Steel, Duncan. 1995. *Rogue Asteroids and Doomsday Comets*. New York: John Wiley & Sons.

Wilson, Jerry D., and Anthony J. Buffa. 1997. *College Physics*. Upper Saddle River, NJ: Prentice Hall.

2000: The Air Traffic Control Problem

Dedicated to the memory of Dr. Robert Machol,
former chief scientist of the Federal Aviation Agency

To improve safety and reduce air traffic controller workload, the Federal Aviation Agency (FAA) is considering adding software to the air traffic control system that would automatically detect potential aircraft flight path conflicts and alert the controller. To that end, an analyst at the FAA has posed the following problems.

- Requirement A: Given two airplanes flying in space, when should the air traffic controller consider the objects to be too close and to require intervention?
- Requirement B: An airspace sector is the section of three-dimensional airspace that one air traffic controller controls. Given any airspace sector, how do we measure how complex it is from an air traffic workload perspective? To what extent is complexity determined by the number of aircraft simultaneously passing through that sector
  - at any one instant?
  - during any given interval of time?
  - during a particular time of day?

  How does the number of potential conflicts arising during those periods affect complexity? Does the presence of additional software tools to automatically predict conflicts and alert the controller reduce or add to this complexity?

In addition to the guidelines for your report, write a summary (no more than two pages) that the FAA analyst can present to Jane Garvey, the FAA Administrator, to defend your conclusions.

Comments

The Outstanding papers were by teams from Duke University, Maggie Walker Governor's School, the U.S. Military Academy, and the University of Colorado. Their papers, together with commentaries, were published in *The UMAP Journal* 21 (3) (2000): 241–310.

Problem Origin

The problem was contributed by Robert Rovinsky (Federal Aviation Agency).

Practitioner's Comments

Jack Clemons (Senior Vice President of Strategic Programs, Lockheed Martin's Air Traffic Company) examined the papers from their relevance to the problems facing real air traffic controllers:

- Thoughtfulness: How well did the team think through the problem statement before addressing it?
- Realism: How close does the proposed solution come to addressing a real-world problem?
- Usefulness: Is the proposed solution itself applicable to address the world of the air traffic controller?

The entry from the University of Colorado team evaluated best. This team spent appreciable time understanding the FAA. The selection of the Denver International Airport as the system model and obtaining the relevant airport, airspace, and ATC parameters from this airport were done extremely well. The team accurately represented the details of the air traffic and correctly asserted that conclusions drawn should translate into other regions. Second, the team used the Federal Aviation Regulations (FARs) as the guidance for resolving technical assumptions. Once again, this shows a real understanding of FAA procedures. FARs are indeed the governing standard for assessing safety.

The use of a "corrected random walk" to validate the FAA's minimum aircraft separation standard was clever. It ignores standard approaches like the computation of minimum maneuver time required for two aircraft approaching head on. However, the model used by this team provides an insight for parallel flight that is novel and confirms the FAA separation standard as well.

The approach taken to quantify stress as a parameter was good. The team researched the available literature to develop a baseline for ATC stress and built the model from there. The model used is both simple and probably applicable. Though arbitrary, the measure of complexity should demonstrate a logical relationship between traffic patterns and workload stress that can provide insight into underlying causes. Of course, the motivation was to allow the team to transform the automation problem into a queueing problem, thus greatly simplifying the model and still providing insights. The simple queueing model that the team proposed overlooked a significant factor in runway access, gate availability—even though all runways are clear, traffic may still have to hold because there are no available passenger gates. However, their model can easily accommodate this factor.

The results are intuitive and appear to be correct. Further, for the problem of airport terminal approach and landing at least, they correctly identify which elements of ATC need remain under the control of humans and which have potential for automation. "I would recommend taking [this paper] forward as is to Ms. Garvey [FAA Administrator] and her staff for review and further evaluation."

Reference

Galotti, Jr., Vincent P. 1998. *The Future Air Traffic Navigation System (FANS): Communications Navigation Surveillance Air Traffic Management*. Ashgate Publishing Company, Ltd.

Machol, Robert E. 1995. Thirty years of modeling midair collisions. *Interfaces* 25 (5) (September-October 1995): 51–72.

Judge's Comments

Many papers recognized that aircraft conflict occurs in pairs and proceeded to assess the maximum number of possible conflicts for a given scenario. Some papers divided the overall airspace into vertically separated layers and then developed conflict algorithms for the 2-D problem on a particular layer as opposed to using a 3-D model from the start. However, several papers fell back on a tacit 3-D model for complexity without realizing that their earlier results did not extend to this case. The most common approach to address the 3-D problem directly was to create an inner collision space (either a sphere or rectangle) around the aircraft and then use a larger alert space (containing this inner collision space) for early warning. The difference in radii between the two spaces was used as a measure for air traffic control conflict reaction time, so that a controller could adjust one of the aircrafts' courses without causing excessive internal forces on the aircraft, its passengers, or its cargo.

There was a wide range of techniques used by teams to represent and identify potential conflicts along aircraft flight paths. Some reduced this problem to a time-parameterized vector-intersection problem in the 2-D plane, and others did the same for the 3-D case as well.

There were several more extensive approaches worth noting:

- One paper assumed that a drift error exists from wind effects, weather, and turbulence along an aircraft flight vector; applied a probability distribution to this error; and constructed a stochastic simulation to test their model.

- Still another paper incorporated both straight and curved parametric flight paths into their methodology. If the flight paths of two aircraft drifted sufficiently close to cause an alert space violation, a controller was presumed to take corrective action. The number of alerts occurring for a given aircraft density in traffic then became a component of complexity measurement.

Teams that dismissed portions of the problem claiming that "FAA ATC conflict software already exists" missed the point: The FAA knows their repertoire of tools; they are seeking other ideas and approaches that might facilitate a better solution than the current one.

You Make the Call: Feasibility of Computerized Aircraft Control

Richard D. Younger
Martin B. Linck
William P. Woessner
University of Colorado
Boulder, CO

Advisor: Anne M. Dougherty

Introduction

We investigate whether some of the work done by air traffic controllers (ATCs) could be handled by computers. Automated systems could act as watchdogs, heading off crises before they become catastrophes. Specifically, we investigate at what point an ATC must take charge of a situation to avoid catastrophe, what sort of decision must be made to remedy the situation, and how much stress is involved.

Objectives

- Define a minimum safe distance between aircraft.
- Develop a numerical model of air traffic around a busy airport.
- Assess system complexity and corresponding ATC workload under a variety of circumstances.
- Develop aircraft guidance algorithms that minimize controller stress.

The System

An ATC has three main tasks as an aircraft approaches an airport, all of which must be carried out as quickly as possible [Wood 1983]:

- Ensure that the aircraft does not collide with another aircraft or any other obstacle.
- Ensure that the approaching craft is inserted smoothly into the traffic around the airport, with a minimum of disruption to the flight paths of other aircraft.

- Guide the aircraft onto a runway, again with a minimum of disruption to the rest of the traffic around the airport.

Special cases, such as aircraft experiencing mechanical malfunctions, medical crises, or fuel shortages, must be dealt with, and changing weather conditions must be taken into account.

To make the simulation concrete, we model Denver International Airport (DIA); our methods could be extended to nearly any air traffic control center.

Each ATC is assigned a specific type of task. Thus, one set of controllers assigns flight paths to incoming aircraft, another guides those aircraft to holding patterns or landing approaches, and yet another guides planes to a safe landing. As an aircraft passes from one controller to another, the pilots switch radio frequencies. Each frequency belongs to a specific controller, and each ATC watches a radar screen on which icons represent flights for which that ATC is responsible. The tower controllers, who are in charge of landings, can see the aircraft and so are not as dependent on radar information [FAA 1999].

At DIA, it is common practice to route all incoming flights on north-south and east-west vectors, since prevailing wind conditions usually favor these approaches [Wood 1983]. Departing flights must use the same runways as incoming flights but once airborne are routed out of DIA airspace on northeast-southeast and northwest-southwest vectors, to minimize conflicts between incoming and outbound aircraft. **Figure 1** shows a diagram of the general approach and departure vectors. To make its final approach and land on a runway, each aircraft has to pass a point in space approximately 5 mi from the end of the landing runway and approximately 2 mi (10,000 ft) above ground level.

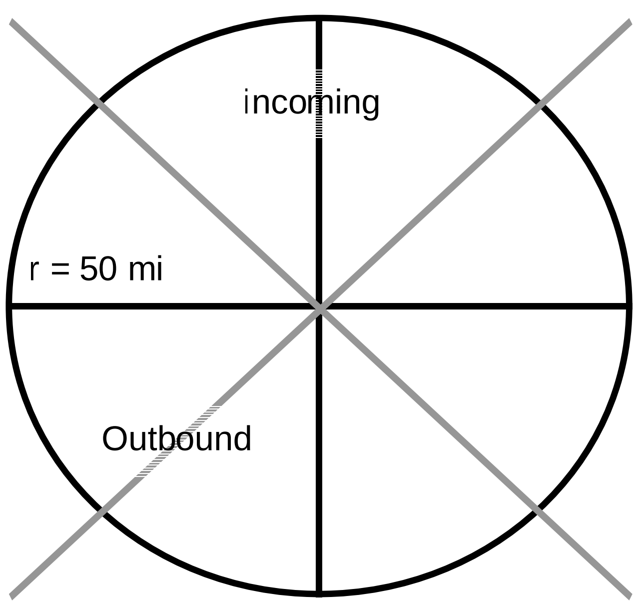


Figure 1. Airspace approach and departure diagram.

Assessing Safety

We use real data wherever possible; we consult the Federal Aviation Regulations (FAR) [FAA 1999] whenever technical questions arose.

Federally regulated Instrument Flight Rules (IFR) state that the minimum safe distance between adjacent aircraft is 1,000 vertical feet and 3 mi of horizontal distance, when aircraft are moving at landing speeds in close proximity to each other and to the airport. Since the runways at DIA are 4,330 ft apart, this minimum safe distance must be ignored on final approach and on runways.

Assessing Complexity, ATC Workload and Stress

Goode and Machol [1957], writing about large-scale queueing systems, describe complexity as "the extent to which any given attribute of a system will affect all the others if it is changed." In a complex system, all the variables are tightly linked; one could not, for example, change the position of an aircraft without immediately having to change some characteristic of most of the other aircraft in the system. Unfortunately, measuring this type of complexity is difficult, since there is no obvious set of measurable system variables for assessing how closely each variable depends on all the others.

Further, we are interested not merely in the complexity of the system but in the stress and fatigue that the system is likely to cause its ATCs. Surprisingly, there is no accepted set of factors that cause ATC stress, fatigue, and error. Some researchers, such as Redding [1992], have concluded that the number of incidents was highest when ATC workload was actually moderate to intermediate. On the other hand, Morrison and Wright [1989], reviewing NASA data, report that ATCs make more mistakes when the workload is at its highest. One innovative study of the phenomenon of ATC fatigue is that by Brookings et al. [1996]. Their test subjects were Air Force ATCs who were asked to play an air traffic control simulation and were exposed to scenarios of varying difficulty. One scenario was an "overload scenario," in which they were asked to coordinate the movements of 15 aircraft at once. As the ATCs attempted to deal with each scenario, their heart rates, blink rates, and brain activity were monitored. Brookings et al. concluded that there was a correlation between workload and operator stress and that the likelihood of error—in the form of separation errors (not enough room between planes), fumbled approaches, and botched handoffs—was directly linked to ATC stress. As a result, we assume for our simulation that ATC stress should be minimized and that there is no "optimal stress level" [Redding 1992] at which ATCs should operate.

Based on our literature survey, we conclude that there are four measurable factors that influence ATC stress levels:

- n, the number of total planes in the airspace around the airport.
- f, the number of separation errors currently occurring in the airspace around the airport. We assume that a single separation error causes as much stress

as 20 extra aircraft in the airspace.

- d, the *smallest* distance between planes currently on the map.
- a, the *average* distance between aircraft. If all the aircraft are well distributed throughout the airspace of the airport, the ATCs are likely to experience less stress than they would if they were all clustered together.

Our formula for the stress-causing complexity experienced by ATCs is

$$C = n + 20f + 100/d + 100/a.$$

Queueing Theory, Stochastic Input, and Algorithm Design

The airport is a queueing system. A queueing system has servers that handle input, perform some function on the input, and then pass the input to some other part of the system. Input is not created or destroyed in the system. The servers are usually referred to as *channels*. The five active runways of DIA are the channels of the system.

When the number of servers is inadequate to the amount of input they are called upon to handle, a queue develops. In the case of an airport, the holding patterns in which aircraft wait for clearance to land are the queues of the system.

When the amount of traffic is stochastic (determined by a probabilistic distribution) and the input is discrete, the input density is often described by a Poisson distribution. The amount of time for each channel to process an input need not be constant. The standard numerical approach to modeling would involve setting up an input generator, which would send us airplanes according to a density function. We would then set up our channels, and the amount of time to process each plane would vary, probably according to a normal distribution. We would also set up holding patterns, to which our planes could be sent when there are no runways available. We could then let the program run and see how queues develop as time passes.

Unfortunately, this simple approach doesn't allow us to investigate all the questions posed by the problem statement. The objective of the simulation should be to develop and test algorithms to monitor the position and speed of aircraft and to alter flight paths to minimize the workload and stress of ATCs. If we treat airplanes simply as inputs, without them becoming objects maneuvering in space, we cannot adequately test those algorithms. So, a more ambitious approach is called for, one that allows us to “create airplanes” stochastically, maneuver them, send them to holding patterns, land them, and assess the value of stress-causing complexity as the simulation runs.

The Model

An airport is a continuous system; each airplane continuously changes position and velocity. However, there are discrete events that characterize the system, such as takeoffs, landings, and handoffs. Both continuous and discrete characteristics of the system can be described by a discrete model provided the time resolution is fine enough. Recognizing this, we develop three numerical simulations of aircraft behavior, which test:

- the minimum-safe-distance assumption,
- guidance algorithms on aircraft landing, and
- guidance algorithms for aircraft entering and maintaining a holding pattern.

Minimum-Safe-Distance Simulation

Consider two aircraft traveling on parallel flight paths at an airspeed of 300 mph, well below the cruising speed of commercial airliners. Let the vertical axis be the z-axis, the axis parallel to the flight path be the x-axis, and the axis perpendicular to both be the y-axis. Wind and other factors perturb the velocity vectors of the planes according to (we assume) a normal distribution with mean zero. A large standard deviation might correspond to the planes flying in heavy weather, a small one to calm weather.

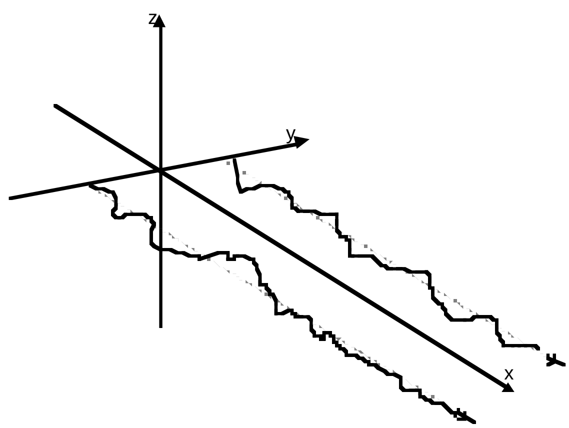


Figure 2. Aircraft separation simulation.

If we did not correct the aircraft's course, the plane would diverge from its flight path and move in a random walk as time passes. Our program determines the distance between the aircraft and its flight path and changes the velocity vector to bring the plane back on course. Airplanes have maximum rates of acceleration in any direction, and we assume that the aircraft can change velocity in any direction by no more than 10 mph/s. The result is a corrected random walk, with the step size normally distributed and a finite correction to each step.

Since the wingspan of an airliner is approximately 200 ft and turbulence effects surround the aircraft, we assume that if the airliners come within 250 ft of one another, they collide. We assume for the sake of simplicity that this holds true in the vertical direction as well. To test our minimum-safe-distance assumption, we fly planes next to each other for 100 h; if the likelihood of collision is less than 0.05%, we consider the distance safe for the given weather conditions.

Developing Aircraft Guidance Algorithms

A flow diagram for our airport is shown in **Figure 3**.

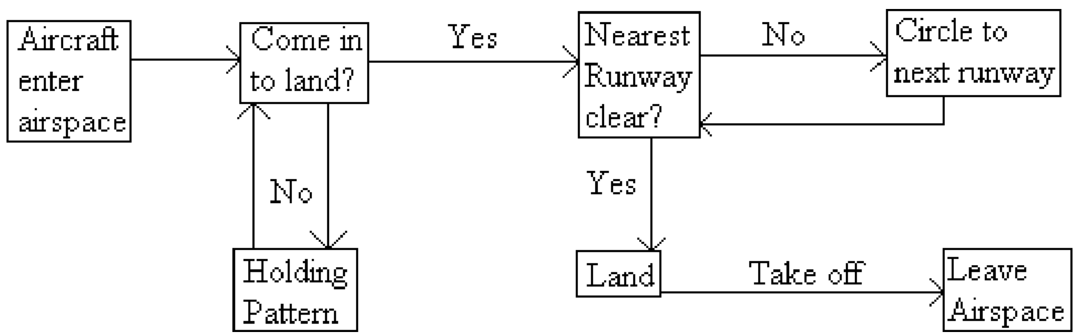


Figure 3. Flow diagram for an airport.

Airplanes enter the airspace according to a stochastic distribution. If a landing approach is free, they proceed towards it. If no runway is clear or if the airspace is too crowded, they are sent to a holding pattern (our queue). The amount of time in the holding pattern depends on the rates at which planes land and planes enter the airspace.

We assume that the scheme used to select the next aircraft cleared for approach to landing is FIFO (first in, first out). If we wished to take into consideration factors such as fuel or other flight emergencies, each plane would have to be assigned a priority and planes would be pulled from the queue according to priority.

Once an aircraft has been cleared for approach, it must select a runway. If the nearest runway is not free, the aircraft must circle until one is. Once it finds a clear runway, it may land. The runway is then occupied for some (perhaps stochastic) amount of time. The aircraft then spends some (perhaps stochastic) amount of time on the ground and takes off again.

The process can be divided into boxes, or areas. How much stress airplanes in a given area cause to the ATCs depends on the number of airplanes and the extent to which guidance algorithms manage to control the aircraft. Aircraft just entering or just leaving the airspace are unstressful, since they are presumably spaced out along a very large circumference. Aircraft within 5 mi of a runway that are coming in to land are admitted only when a runway is clear; they are handled by ATCs who monitor those flights visually as well as via radar. The consequences of a mistake in this function are so high that we can safely assume that humans will handle this task for the foreseeable future.

There remain two areas in which our algorithms might ease controller stress:

- aircraft that have received clearance and must line up for a landing. **Figure 4** shows runway checkpoints, through one of which an aircraft must pass to land.

- aircraft that have not been cleared to land and must be sent to a queue. **Figure 5** shows the set of checkpoints that comprise the holding pattern. An aircraft can enter the holding pattern at any checkpoint, and, if properly guided, proceeds to the next checkpoint in the sequence. It cycles through the sequence until given clearance to approach. All aircraft move through the sequence in the same direction, to avoid head-on collisions.

We design algorithms to maintain aircraft spacing while guiding those aircraft, either toward a runway checkpoint, or through the circular set of checkpoints that form the queue.

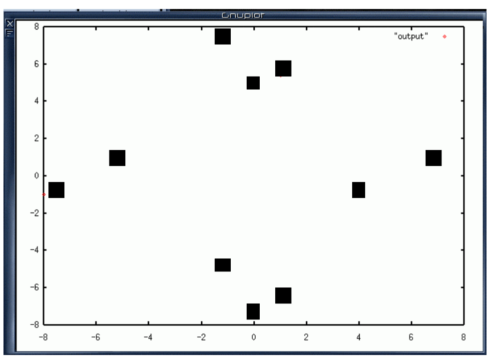


Figure 4. Runway checkpoints.

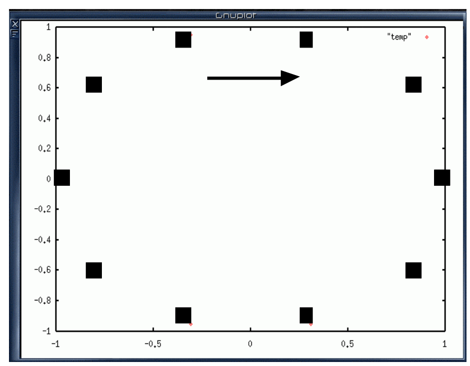


Figure 5. Queue checkpoints. Arrow indicates flight direction.

The Algorithms

Single-Avoidance

This algorithm determines the distance between each aircraft and that aircraft's next checkpoint and orients the aircraft toward the checkpoint. It also evaluates the distance between that aircraft all other aircraft; if any distance is equal to or smaller than the minimum distance (3 mi horizontal, 1,000 ft vertical), it then orients the aircraft directly away from the dangerously close airplane (the aircraft in the airspace longer changes course) without changing speed. Once the distance again exceeds the minimum safe distance, the aircraft that had to change its course looks for its nearest objective (which need not be the same objective that it was originally approaching) and changes course toward that objective. An example of aircraft being guided toward a landing checkpoint by the single-avoidance algorithm is shown in **Figure 6**.

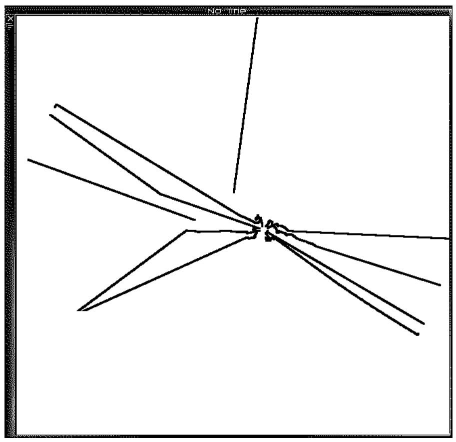


Figure 6. Aircraft converging on runway checkpoints while guided by the Single-Avoidance Algorithm.

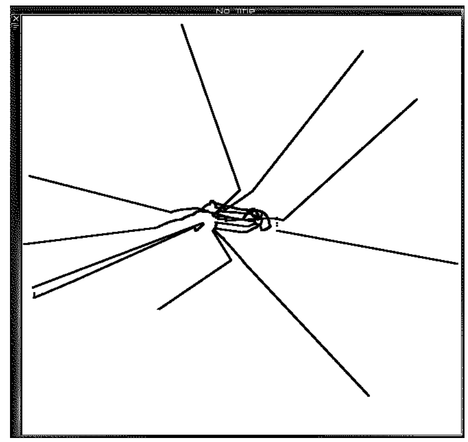


Figure 7. Aircraft moving toward landing checkpoints under the Double-Avoidance Algorithm.

Double Avoidance

If the distance between two aircraft drops below the minimum safe distance, *both* aircraft head away from each other, changing courses equally and in opposite directions without changing speeds. **Figure 7** shows an aircraft being guided toward a landing checkpoint by the Double-Avoidance Algorithm.

Single-Vector Repulsion

Unlike the first two algorithms, this algorithm constantly measures the distance between the aircraft under its control and that aircraft's nearest neighbor. It alters the course of that aircraft before a separation error can occur. The scheme used to correct the aircraft's flight path is as follows.

In each time step, the algorithm determines the direction in which the aircraft must fly to reach the nearest checkpoint. It thus establishes a velocity vector $\vec{V}$ for the aircraft, whose magnitude cannot change but whose direction is corrected in every time step. It then finds the vector $\vec{D}$ that connects the craft under guidance with its nearest neighbor and calculates a correction vector $\vec{A}$ whose direction is the same as that of $\vec{D}$ but whose magnitude is

$$||\vec{A}|| = \frac{b}{||\vec{D}||^2},$$

where b is a scaling constant. If b is large, the correction is severe, even at large distances; if b is very small, we run the risk that flight paths will not be adjusted quickly enough and the aircraft will come to close to each other.

The velocity vector $\vec{V}$ is corrected to $\vec{V_c} = \vec{V} - \vec{A}$. The result is that every aircraft is repelled by the aircraft nearest to it, with increasing intensity as the distance between the adjacent aircraft becomes smaller. At the same time, the aircraft remains attracted to its objective.

If there are only two aircraft in the simulation, and both are headed for the same objective, they behave like a pair of negatively charged ions approaching a large positively charged sphere (this analogy breaks down when there are more than two aircraft). **Figure 8** shows aircraft maneuvering toward an objective while guided by the Single-Vector Repulsion Algorithm. Note that most of the aircraft maintain much better spacing than with the previous algorithms.

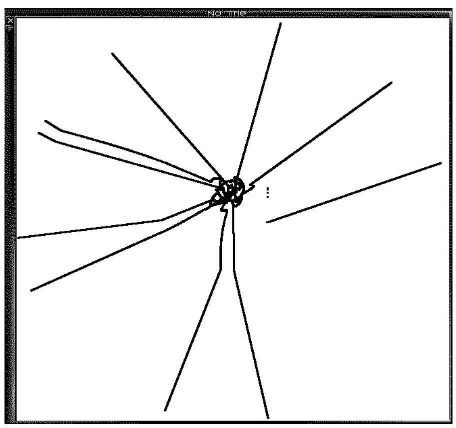


Figure 8. Aircraft maneuvering toward landing checkpoints under the guidance of the Single-Vector Repulsion Algorithm.

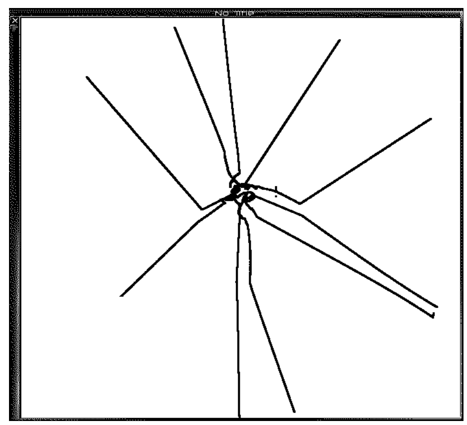


Figure 9. The Multiple-Vector Repulsion Algorithm.

Multiple-Vector Repulsion

The Multiple-Vector Repulsion Algorithm has the same vector mechanics as the Single-Vector Repulsion Algorithm, but each airplane is repelled not just by its nearest neighbor but by every other airplane in the airspace. The behavior of a number of aircraft headed toward a single objective would be roughly analogous to a group of negatively charged ions heading toward a large positively charged sphere. With more than one objective, however, the analogy is less apt, since each "ion" is attracted only to the nearest sphere. **Figure 9** shows 10 aircraft approaching a checkpoint under the guidance of the Multiple-Vector Repulsion Algorithm.

Testing the Algorithms

The Landing Approach Test

In this test, aircraft enter the airspace according to a normal distribution with mean 120 s and a standard deviation of 60 s. The points on the circumference of the airspace are evenly and randomly distributed. The airspace is centered on the airport and has a radius 50 mi. Each aircraft enters the airspace at an altitude of 6 mi (close to cruising altitude) and descends to one of 10 runway checkpoints (each of the 5 runways can be approached from either end), each located 5 mi from the end of a runway and at an elevation of 2 mi. Once an airplane reaches a runway checkpoint, its velocity instantly becomes zero. This ensures that no further aircraft can reach that checkpoint while the plane remains in place. After a set period of time (1 min in our simulation, based on the FAR [FAA 1999]), the airplane disappears and can be described as having landed. The runway is then clear and a new aircraft can occupy that checkpoint. The value for ATC stress is calculated at each time step.

The Queueing Test

Aircraft enter the airspace just as they do in the landing approach test. They descend to a set of queueing checkpoints and maneuver around those checkpoints in a holding pattern. Each holding pattern has a saturation level of airplanes, equal to the perimeter of the pattern divided by twice the minimum allowable distance between aircraft. Aircraft are added to the holding pattern stochastically until saturation is reached. We calculate the amount of ATC stress that the holding pattern generates. **Figure 10** shows 10 aircraft descending toward and entering the holding pattern under the control of the Single-Vector Repulsion Algorithm.

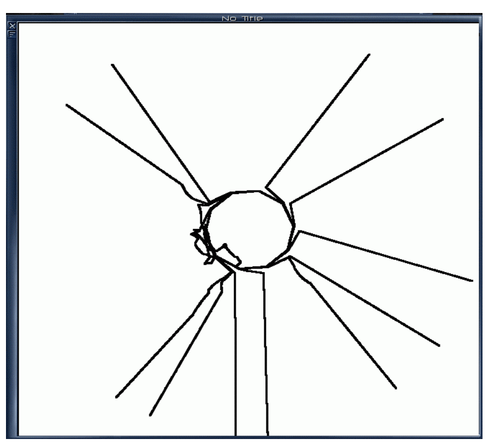


Figure 10. Aircraft being maneuvered in to a holding pattern by the Single-Vector Repulsion Algorithm.

Simplifying Assumptions in Testing the Algorithms

- All aircraft behave like commercial air carrier aircraft.
- The velocity of all aircraft is 300 mph.
- The minimum vertical separation between aircraft is 1,000 ft.
- The minimum horizontal separation between aircraft is 3 mi.
- The turning radius of all aircraft is negligibly small compared to the overall airspace.
- An aircraft has reached a checkpoint when it has passed within 1 mi of that checkpoint.
- Weather conditions do not affect the behavior of aircraft and are not considered.
- No aircraft is given any special priority over any other; fuel and emergency considerations are therefore ignored.
- There is no coordinating intelligence at work. The human ATCs can watch the simulation (and be stressed by it), but all actions of the aircraft are determined by the algorithm being tested.
- Aircraft are incapable of acting without instructions from the control algorithms. In essence, the pilot of each aircraft blindly and unquestioningly follows the orders given by the algorithm.

The following assumptions apply only to the Landing Approach test for all algorithms:

- A runway approach checkpoint remains occupied for 1 min.
- Any aircraft that reaches an approach checkpoint lands; there are no failed landing attempts.
- Once an aircraft has landed, it ceases to interact with any other aircraft and disappears from the simulation.
- Outbound aircraft do not interact with inbound aircraft and hence do not appear in the simulation. **Figure 11** shows that with our current checkpoint system, corridors develop in which there is no inbound traffic. So we assume that outbound traffic passes through these corridors and need not be accounted for.

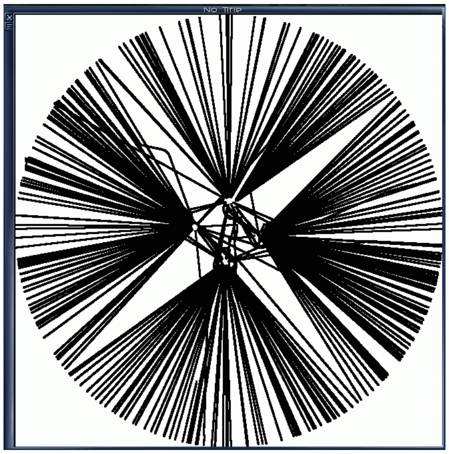


Figure 11. Typical traffic pattern when aircraft head toward the checkpoints from the periphery. Note the open corridors, along which outbound traffic can be routed.

Results and Commentary

Minimum-Safe Distance Simulation

Figure 2 shows the flight paths of two aircraft as they travel next to each other for 5 min, together with ideal flight paths that are separated by 3 mi. The results of many such simulations are given in **Figure 12**. Each data point represents the average of 150 runs, with each run lasting 100 h. The components of each aircraft's velocity are each disturbed by components with a mean of zero and with standard deviations as noted in the legend. The disturbances correspond to moderate, heavy, and severe weather conditions.

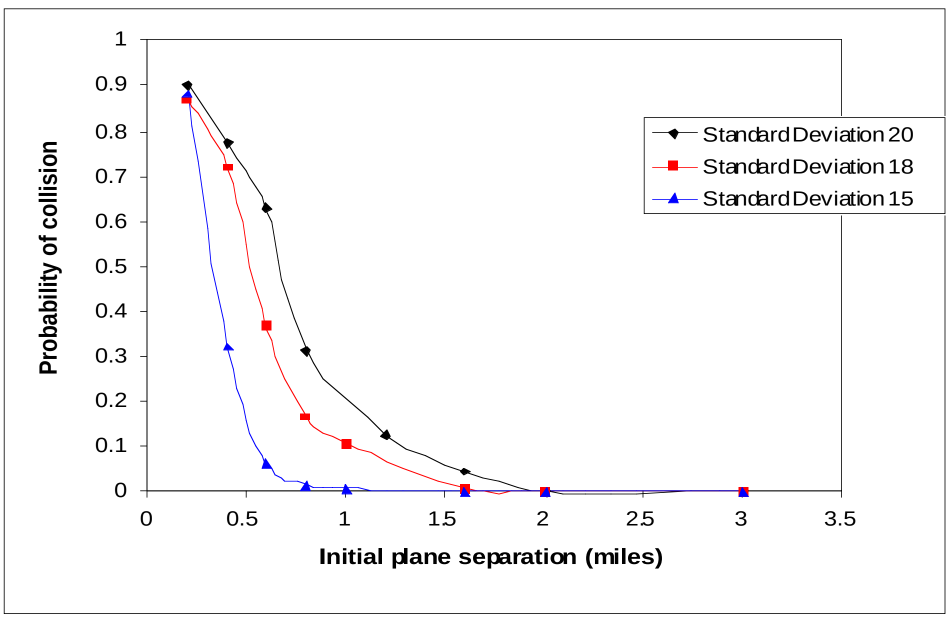


Figure 12. Probability of collision in a 100 h run as a function of initial separation.

The likelihood of collision is well under our 0.05% criterion when the separation distance is 3 mi. In fact, in the course of all 450 runs of 100 h, over all three disturbance levels, no collisions occur. The likelihood of collision increases exponentially as the horizontal separation distance decreases and becomes appreciable at 1.8 mi, where the first collision occurs. We conclude that FAA regulations give a secure safety margin and so abide by them for the rest of the simulation.

Holding Pattern and Landing Approach Simulations

There are no statistically significant differences among the stress levels for our four test algorithms, neither for holding pattern nor landing approach simulations. The holding pattern simulation achieves equilibrium with little to no stress and then the stress levels out. Since the landing simulation is much more complicated, one would expect sharp spikes and peaks in the stress, and this is what we see. **Figures 13** (generated using the Single-Vector Repulsion Algorithm) and **14** (generated using the Multiple-Vector Repulsion Algorithm) show sample graphs of stress vs. time for the queueing and landing simulations, respectively.

In **Figure 13**, the stress rises as planes are added; when the pattern is saturated, the stress level becomes steady. There are few collision warnings; the rise in stress is caused by the decreasing proximity of the planes as they approach the holding pattern.

Figure 14, however, is fascinating. For $0 < t < 5{,}000$, the graph is piecewise continuous, but the rest is totally discontinuous. In our definition of stress, there are two continuous terms and two discrete terms. Over the first 5,000 s of the simulation, the continuous terms play an important role in stress; but during the latter 5,000 s, the continuous terms die off, leaving the discrete terms to determine the stress.

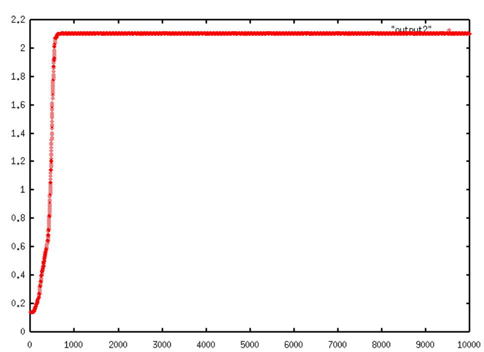


Figure 13. Stress vs. time for a queueing simulation.

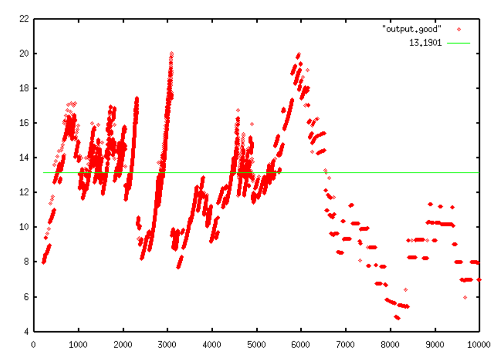


Figure 14. Stress vs. time for a landing simulation.

Table 1 gives summary statistics for both simulations run with each of the controlling algorithms. Each scenario was repeated 100 times. The seed for the pseudorandom number generator was held constant across the controlling algorithms, so that each algorithm received the same input.

Table 1.

Descriptive statistics from the simulations.

	Landing Simulation		Queueing Simulation	
	Mean	SD	Mean	SD
Single Avoidance	12.8	0.8	4.1	1.9
Double Avoidance	14.2	1.7	4.1	1.9
Single-Vector Repulsion	14.2	1.7	4.1	1.9
Multiple-Vector Repulsion	13.2	1.6	4.8	2.2

The most interesting result is that the Single Avoidance Algorithm performs better than the other more clever algorithms. Here we can draw a parallel to the phenomenon from computer science known as *deadlock*. Deadlock occurs when two or more processes cannot continue executing because each process is requesting resources owned by the other process [Nutt 1999]. Despite much research and many clever algorithms, the standard way to handle deadlock in a computing environment is to just pick a winner, usually the process that has been waiting the longest. The Single Avoidance Algorithm is analogous: When two planes request the same airspace, only one can be awarded it; the algorithm picks a winner and vectors the loser away.

To make the Multiple-Vector Repulsion Algorithm more efficient, we increase the repulsion factor between planes; if planes are kept farther apart, they will not incur collision warnings. The idea is good but the results are not encouraging. **Figure 15** is a plot of stress vs. time for the landing simulation using the Multiple-Vector Repulsion Algorithm with increased repulsion.

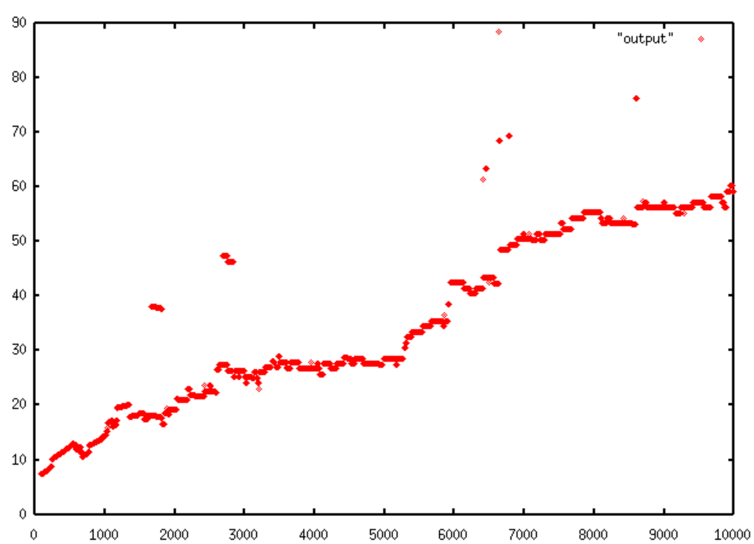


Figure 15. Stress vs. time for the landing simulation using the Multiple-Vector Repulsion Algorithm with increased repulsion.

We see a monotonically increasing stress level—certainly not the desired result. We can explain this phenomenon too in terms of deadlock. Imagine that two airplanes approach a checkpoint on opposite approach vectors. When they get close enough to each other, the repulsion factor makes them turn around. When they again get far enough away from each other, they turn around and fly back toward the checkpoint; and the cycle repeats. The planes become deadlocked, so very few planes can land. Hence, the number of planes in the airspace increases, leading to increasing stress.

The queueing simulations generate very little stress. Hence, software should be able to take control of a plane, vector it into a holding pattern, and keep it there.

Strengths and Weaknesses

More factors contribute to this system than we are able to consider in a weekend. However, our model is modularized to the point where modules can be run independently of each other, making it easy to focus on specific parts of the model.

It took our workstation approximately two hours to generate the data for our limited model, but parallel processors could each handle a set of planes.

Our model covers all of the major elements of an airport simulation; while it is based on DIA, it can easily be generalized.

Conclusions and Recommendations

We conclude from our simulation that *FAA guidelines for aircraft separation give a generous margin* for navigational, technical, and pilot error. Perhaps they could be relaxed, especially for well-controlled parallel landing situations in good to moderate weather.

The air traffic control problem exhibits many traits common to NP-complete problems:

- There is no deterministic way to pick the best solution.
- Humans appear to control air traffic much more effectively than computers could.
- Changes in the control algorithm for our simulation do not seem to impact the quality of the solution at all.

These features lead us to speculate that this problem is indeed NP-complete.

The computer does, however, handle the queueing problem fairly well; planes are vectored into the queue and held there with minimal stress. These results are promising; they suggest that *though computers should not replace humans as the primary means of air traffic control, computers might be capable of handling the more mundane tasks.*

References

Airports Data Base — Denver International Airport. 2000. `http://www.boeing.com/assocproducts/noise/denver.html` .

Brookings, J.B., G.F. Wilson, and C.R. Swain, 1996. Psychophysiological responses to changes in workload during simulated air traffic control. *Biological Psychology* 42: 361–77.

Federal Aviation Administration. 1999. *Federal Aviation Regulations.* Washington, DC: Government Printing Office.

Goode, Harry H., and Robert E. Machol. 1957. *System Engineering.* New York: McGraw-Hill.

Morrison, R., and R.H. Wright. 1989. ATC control and communication problems: An overview of recent ASRS data. In *Proceedings of the Fifth International Symposium on Aviation Psychology*, 902–907.

Nutt, Gary. 1999. *Operating Systems: A Modern Perspective*. Reading, MA: Addison-Wesley.

Redding, R.E. 1992. Analysis of operational errors and workload in air traffic control. In *Proceedings of the 36th Annual Meeting of the Human Factors Society*, 1321–1325.

Wood, Kenneth Ray. 1983. Airport activity simulation. Master of Science thesis, University of Colorado. Boulder, CO.

2001: The Hurricane Evacuation Problem

Evacuating the coast of South Carolina ahead of the predicted landfall of Hurricane Floyd in 1999 led to a monumental traffic jam. Traffic slowed to a standstill on Interstate I-26, which is the principal route going inland from Charleston to the relatively safe haven of Columbia in the center of the state. What is normally an easy two-hour drive took up to 18 hours to complete. Many cars simply ran out of gas along the way. Fortunately, Floyd turned north and spared the state this time, but the public outcry is forcing state officials to find ways to avoid a repeat of this traffic nightmare.

The principal proposal put forth to deal with this problem is the reversal of traffic on I-26, so that both sides, including the coastal-bound lanes, have traffic headed inland from Charleston to Columbia. Plans to carry this out have been prepared (and posted on the Web) by the South Carolina Emergency Preparedness Division. Traffic reversal on principal roads leading inland from Myrtle Beach and Hilton Head is also planned.

A simplified map of South Carolina is shown in **Figure 2**. Charleston has approximately 500,000 people, Myrtle Beach has about 200,000 people, and another 250,000 people are spread out along the rest of the coastal strip. (More accurate data, if sought, are widely available.)

The interstates have two lanes of traffic in each direction except in the metropolitan areas, where they have three. Columbia, another metro area of around 500,000 people, does not have sufficient hotel space to accommodate the evacuees (including some coming from farther north by other routes); so some traffic continues outbound on I-26 towards Spartanburg, on I-77 north to Charlotte, and on I-20 east to Atlanta. In 1999, traffic leaving Columbia going northwest was moving only very slowly.

Construct a model for the problem to investigate what strategies may reduce the congestion observed in 1999. Here are the questions that need to be addressed:

1. Under what conditions does the plan for turning the two coastal-bound lanes of I-26 into two lanes of Columbia-bound traffic, essentially turning the entire I-26 into one-way traffic, significantly improve evacuation traffic flow?

2. In 1999, the simultaneous evacuation of the state's entire coastal region was ordered. Would the evacuation traffic flow improve under an alternative strategy that staggers the evacuation, perhaps county by county over some time period consistent with the pattern of how hurricanes affect the coast?

3. Several smaller highways besides I-26 extend inland from the coast. Under what conditions would it improve evacuation flow to turn around traffic on these?

Figure 2. Highways in South Carolina.

4. What effect would it have on evacuation flow to establish additional temporary shelters in Columbia, to reduce the traffic leaving Columbia?

5. In 1999, many families leaving the coast brought along their boats, campers, and motor homes. Many drove all of their cars. Under what conditions should there be restrictions on vehicle types or numbers of vehicles brought in order to guarantee timely evacuation?

6. It has been suggested that in 1999 some of the coastal residents of Georgia and Florida, who were fleeing the earlier predicted landfalls of Hurricane Floyd to the south, came up I-95 and compounded the traffic problems. How big an impact can they have on the evacuation traffic flow?

Clearly identify the measures of performance that are used to compare strategies.

Required: Prepare a short newspaper article, not to exceed two pages, explaining the results and conclusions of your study to the public.

Comments

The Outstanding papers were by teams from Bethel College, Duke University, Maggie Walker Governor's School, Harvey Mudd College, Lawrence Technological University, and Wake Forest University. Their papers, together with a judge's commentary, were published in *The UMAP Journal* 22 (3) (2001): 257–343.

Problem Origin

The problem was contributed by Jerrold R. Griggs (University of South Carolina).

Judge's Comments

Mark Parker (Carroll College, MT) divided the teams' approaches into two general categories:

Macroscopic: Traffic on a particular highway or segment of highway was considered to be a stream, and a flow rate for the stream was characterized. Among the successful approaches in this category were fluid dynamics and network flow algorithms.

Microscopic: These can be considered car-following models, where the spacing between and the speeds of individual vehicles are used to determine the flow. Among the successful approaches were discrete event simulations (including cellular automata) and queuing systems.

By far, the most common approach was to determine that the flow q, or flux, is a function of the density ρ of cars on a highway and the average speed v of those cars: $q = \rho v$. Successful approaches identified the following characteristics of the basic traffic flow problem:

- When the vehicle density on the highway is 0, the flow is also 0.
- As density increases, the flow also increases (up to a point).
- When the density reaches its maximum, or *jam density* ρ_0, the flow must be 0.
- Therefore, the flow initially increases, as density does, until it reaches some maximum value. Further increase in the density, up to the jam density, results in a reduction of the flow.

At this point, many teams either derived from first principles or used one of the many resources available on traffic modeling (such as Garber and Hoel [1999]) to find a relationship between the density and the average speed. Three of the common macroscopic models were:

- a linear model developed by Greenshield:

$$v = v_0 \left(1 - \frac{\rho}{\rho_0}\right), \qquad \text{so} \qquad q = \rho v_0 \left(1 - \frac{\rho}{\rho_0}\right);$$

- a fluid-flow model developed by Greenberg:

$$v = v_0 \log \frac{\rho}{\rho_0}, \qquad \text{so} \qquad q = \rho v_0 \log \frac{\rho}{\rho_0};$$

  or

- a higher-order model developed by Jayakrishnan:

$$v = v_0 \left(1 - \frac{\rho}{\rho_0}\right)^a, \qquad \text{so} \qquad q = \rho v_0 \left(1 - \frac{\rho}{\rho_0}\right)^a,$$

where v_0 represents the speed that a vehicle would travel in the absence of other traffic (the speed limit). By taking the derivative of the flow equation with respect to speed (or density), teams then found the optimal speed (or density) to maximize flow.

Many teams took the optimal flow from one of the macroscopic approaches and used it as the basis for a larger model. One of the more common models was simulation, to determine evacuation times under a variety of scenarios.

To make it beyond the Successful Participant category, teams had to find a way *realistically* to regulate traffic density to meet these optimality conditions. Many teams did this by stipulating that ramp metering systems (long term) or staggered evacuations (short term) could be used to control traffic density.

A number of mathematically rigorous papers started with a partial differential equation, derived one of the macroscopic formulas, determined appropriate values for the constants, calculated the density giving the optimal flow, and incorporated this flow value into an algorithm for determining evacuation time. In spite of the impressive mathematics, if no plan was given to regulate traffic density, the team missed an important concept of the MCM: the realistic application of a mathematical solution to a real-world problem.

To survive the cut from Honorable Mention to Meritorious, a paper had to have a unique aspect on some portion of the problem, such as a unique modeling approach or some aspect of the problem analyzed particularly well. Thus, papers that failed to address all questions or had a fatal weakness that prevented their model from being extended could still be considered Meritorious. The Meritorious papers typically had very good insight into the problem but deficiencies as minor as missing parameter descriptions or model implementation details.

To be Outstanding, a paper had to meet the minimum requirements of:

- addressing all 6 questions,
- including all required elements (e.g., the newspaper article), and

- including some sort of validation of the model.

We were also particularly interested in how teams modeled the I-26 / I-95 interchange and the congestion problem in Columbia. Many teams chose to treat Columbia as the terminal point of their model and assumed that all cars arriving there would be absorbed without creating backups.

The six Outstanding papers:

- developed a solid model that allowed them to address all six questions, and analyze at least one very thoroughly;
- made a set of clear recommendations;
- analyzed their recommendations within the context of the problem; and
- wrote a clear and coherent paper describing the problem, their model, and their recommendations.

The INFORMS prizewinner, from Lawrence Technical University, combined Greenshield's model with a discrete event simulation: a solid paper with logical explanations and good analysis. The team's model handled bottlenecks, and the team used a simulation of the actual 1999 evacuation to validate their model.

Blowin' in the Wind

Mark Wagner
Kenneth Kopp
William E. Kolasa
Lawrence Technological University
Southfield, MI

Advisor: Ruth G. Favro

Introduction

We present a model to determine the optimal evacuation plan for coastal South Carolina in the event of a large hurricane. The model simulates the flow of traffic on major roads. We explored several possible evacuation plans, comparing the time each requires.

Traffic flow can be significantly improved by reversing the eastbound lanes of I-26 from Charleston to Columbia. By closing the interchange between I-26 and I-95 and restricting access to I-26 at Charleston, we can reduce the overall evacuation time from an original 31 h to 13 h.

However, a staggered evacuation plan, which evacuates the coastline county by county, does not improve the evacuation time, since traffic from each coastal population center interferes little with traffic flowing from other areas being evacuated. Although reversing traffic on other highways could slightly improve traffic flow, it would be impractical. Restrictions on the number and types of vehicles could speed up the evacuation but would likely cause more problems than improvements.

Theory of Traffic Flow

We require a model that simulates traffic flow on a large scale rather than individual car movement. We take formulas to model traffic flow from Beltrami [1998]. Although traffic is not evenly distributed along a segment of road, it can be modeled as if it were when large segments of road are being considered. We can measure the traffic density of a section of road in cars / mi. The traffic speed u at a point on the road can be calculated from the density according to the formula

$$u(r) = u_m \left(1 - \frac{\rho}{\rho_m}\right),$$

where ρ is the traffic density, u_m is the maximum speed of any car on the road, and ρ_m is the maximum traffic density (with no space between cars). We define the *flow* of traffic at a point on the road as the number of cars passing that point in a unit of time. The flow q can be easily calculated as

$$q(\rho) = \rho u.$$

It is the flow of traffic that we desire to optimize, since greater flow results in a greater volume of traffic moving along a road.

Assumptions

- During an evacuation, there is an average of 3 people per car. This is reasonable, since people evacuate with their entire families, and the average household in South Carolina has 2.7 people, according to the 1990 census.
- The average length of a car on the road is about 16 ft.
- In a traffic jam, there is an average of 1 ft of space between cars.
- The two above assumptions lead to a maximum traffic density of

$$\frac{5280 \text{ ft / mile}}{17 \text{ ft / car}} = 310 \text{ cars / mile / lane}.$$

- The maximum speed is 60 mph on a 4-lane divided highway, 50 mph on a 2-lane undivided country road.
- Vehicles follow natural human tendencies in choosing directions at intersections, such as preferring larger highways and direct routes.
- The traffic flow of evacuees from Florida and Georgia on I-95 is a continuous stream inward to South Carolina.
- When vehicles leave the area of the model, they are considered safely evacuated and no longer need to be tracked.
- There will not be traffic backups on the interstates at the points at which they leave the area of the model.
- A maximum of 30 cars / min can enter or exit a 1-mi stretch of road in a populated area, by means of ramps or other access roads. Up to the maximum exit rate, all cars desiring to exit a highway successfully exit.
- The weather does not affect traffic speeds. The justifications are:
  - During the early part of the evacuation, when the hurricane is far from the coast, there is no weather to interfere with traffic flowing at the maximum speed possible.

  - During the later part of the evacuation, when the hurricane is approaching the coast, traffic flows sufficiently slowly that storm weather would not further reduce the speed of traffic.

- There is sufficient personnel available for any reasonable tasks.

Objective Statement

We measure the success of an evacuation plan by its ability to evacuate all lives from the endangered areas to safe areas between announcement of mandatory evacuation and landfall of the hurricane; the best evacuation plan takes the shortest time.

Model Design

The Traffic Simulator

Our traffic simulator is based on the formulas above. Both space and time are discretized, so that the roads are divided into 1-mi segments and time is divided into 1-min intervals. Vehicles enter roads at on-ramps in populated areas, leave them by off-ramps, and travel through intersections to other roads.

Each 1-mi road segment has a density (the number of cars on that segment), a speed (mph), and a flow (the maximum number of cars that move to the next 1-mile segment in 1 min). Each complete road section has a theoretical maximum density ρ_m and a practical maximum density ρ'_m (accounting for 1 ft of space between cars), which can never be exceeded.

Moving Traffic Along a Single Road

The flow for each road segment is calculated as

$$q(\rho) = \frac{\rho u}{u_m}.$$

If the following road segment is unable to accommodate this many cars, the flow is the maximum number of cars that can move to the next segment.

Moving Traffic Through Intersections

When traffic reaches the end of a section of road and arrives at an intersection, it must be divided among the exits of the intersection. For each intersection, we make assumptions about percentages of cars taking each direction, based on the known road network, the capacities of the roads, and natural human tendencies. If a road ends at an intersection with no roads leading out (i.e., the state border), there is assumed to be no traffic backup; traffic flow simply continues at the highest rate possible, and the simulation keeps track of the number of cars that have left the model.

Conflicts occur when more cars attempt to enter a road section at an intersection than that road section can accommodate. Consider a section of road that begins at an intersection. Let:

$q_{\max} = \rho'_m - \rho$ = the maximum influx of cars the road can accommodate at the intersection,

$q_1, \ldots, q_n$ = the flows of cars entering the road at an intersection, and

$q_{\text{in}} = \sum q_i$ = the total flow of cars attempting to enter the road at the intersection.

If $q_{\text{in}} > q_{\max}$, then we adjust the flow of cars entering the road from its entrance roads as follows:

$$q'_i = \frac{q_i}{q_{in}}\, q_{\max}.$$

Therefore, q'_i is the number of cars entering the road from road i. The flow of traffic allowed in from each road is distributed according to the flow trying to enter from each road. Clearly, $\sum q'_i = q_{\max}$.

Simulating Populated Areas

A section of road that passes through a populated area has cars enter and leave by ramps or other access roads. We assume that the maximum flow of traffic for an access ramp is 30 cars / min. We estimate the actual number of cars entering and leaving each road segment based on the population of the area.

Cars cannot enter a road if its maximum density has been reached. For simplicity, however, we assume that cars desiring to exit always can, up to the maximum flow of 30 cars / min per exit ramp.

We desire to know how the population of each populated area changes during the evacuation, so that we can determine the time required. Therefore, we keep track of the population in the areas being evacuated, Columbia, and certain other cities in South Carolina. If all people have been evacuated from an area, no more enter the road system from that area.

Areas do not have to be evacuated immediately when the simulation starts. Each area may be assigned an evacuation delay, during which normal traffic is simulated. Once the delay has passed, traffic in the area assumes its evacuation behavior.

Completing an Evacuation

The six coastal counties of South Carolina (where Charleston includes the entire Charleston area) and the roads leading inland from these areas must be evacuated. When the population of these areas reaches zero, and the average traffic density along the roads is less than 5 cars / mi, the evacuation is complete and the simulation terminates.

Implementing the Model

We implemented the model described above in a computer program written in C++. The logic for the main function is as follows: For each road, we let traffic exit, resolve traffic at intersections, move traffic along the rest of the road, and finally let cars enter the road. We loop until the evacuation is complete.

Traffic flow is considered simultaneous; the traffic flow along every road is determined before traffic densities are updated. However, exits occur first and entrances last, to accurately simulate traffic at access ramps.

Model Results

Simulating the 1999 Evacuation

To simulate the evacuation of 1999, we prepared a simplified map that includes the interstates, other the 4-lane divided highways, and some 2-lane undivided roads. We simulated the evacuation of the coastal counties—Beaufort, Jasper, Colleton, Georgetown, and Horry (including Myrtle Beach)—and the Charleston metro area. The inland areas we considered are Columbia, Spartanburg, Greenville, Augusta, Florence, and Sumter. In addition, we simulated large amounts of traffic from farther south entering I-95 N from the Savannah area. A map of the entire simulation is shown in **Figure 1**.

The results of running this simulation with conditions similar to those of the actual evacuation produced an evacuation time of 31 h to get everyone farther inland than I-95. This is significantly greater than the actual evacuation time and completely unacceptable. The increase in time can be explained by two features of the actual evacuation that are missing in the simulation:

- Only 64% of the population of Charleston left when the mandatory evacuation was announced [Cutter and Dow 2000; Cutter et al. 2000]; our model assumes that everyone leaves.

- Late in the day, the eastbound lanes of I-26 were reversed, eliminating the congestion.

Simulating Reversal of I-26

In this simulation, I-26 E was turned into a second 2-lane highway leading from Charleston to Columbia. The evacuation time was reduced to 19 h. Under

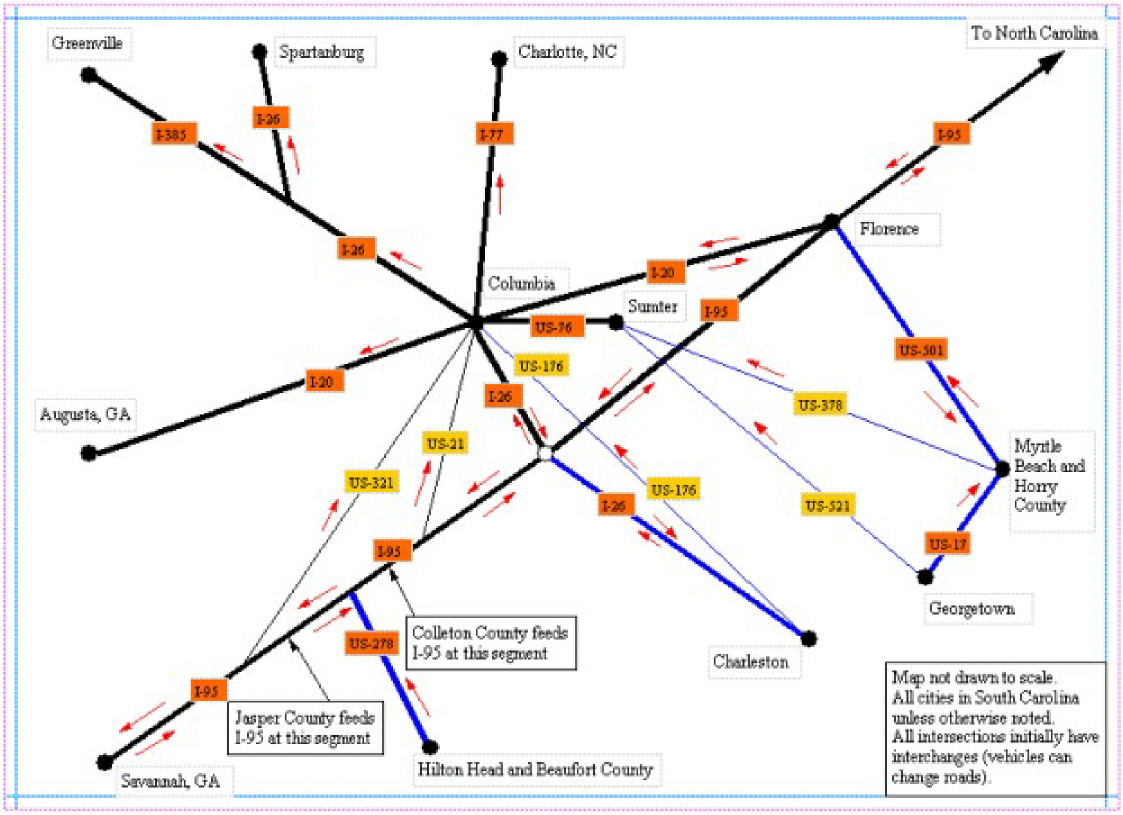


Figure 1. Map of the simulation.

all conditions tested, reversing traffic on the eastbound lanes of I-26 significantly reduces evacuation time.

Simulating a Staggered Evacuation

A staggered evacuation of the coastal counties of South Carolina, going from south to north with 1 h delays, decreases the time for evacuation to 15.5 h—2.5 longer than the best time (described below). This is because the second-slowest county to evacuate, Horry County, is the northernmost and the last to evacuate. An analysis of the evacuation routes used reveals why there is no improvement: The roads for the large counties do not intersect until they reach Columbia. Given that the evacuation of Charleston County takes 13 h, the evacuations of the other large counties (Horry and Beaufort) would need to be advanced or delayed at least this much to have any effect.

Reversing Other Highways

Reversing traffic on smaller highways might improve traffic flow, but this is not a practical option. None of the roads besides I-26 is a controlled-access road; therefore, it is impossible to ensure that the traffic entering the reversed lanes would all move in the desired direction. A single vehicle entering and attempting to travel in the undesired direction would cause a massive jam.

The possible minor highways to Columbia that could be reversed are U.S. highways 321, 176, 521, 378, 501, and 21. All are non-controlled-access roads, meaning that there are no restrictions on where vehicles may exit or enter. Together, they have 450 mi of roadway. A quick examination of U.S. 501, the highest-capacity of these, reveals two intersections per mile with other roads. Considering this as typical, there are 900 intersections outside of towns that would need to be blocked. Factoring in the no fewer than 60 towns along the way, the blocking becomes prohibitive.

Therefore, reversal of minor highways leading inland is not feasible. The only road that can be feasibly reversed is I-26.

Adding Temporary Shelters to Columbia

According to our simulation, the population of the Columbia area after the evacuation (in the best-case scenario) was 1,147,000, a massive number above the 516,000 permanent residents. If more temporary shelters were established in Columbia, there would be less traffic leaving the city and therefore more congestion within the city. This would reduce the rate at which traffic could enter Columbia and lead to extra traffic problems on the highways leading into it. The effect of this congestion is beyond our computer simulation.

We investigated buildings for sheltering evacuees. Using `smartpages.com` to search for schools, hotels, and churches in the Columbia area, we found the numbers of buildings given in **Table 1**. We assumed an average capacity for each type of building. According to the table, Columbia can shelter 1,058,251; this leaves a deficit of 89,000.

Table 1.

Post-evacuation sheltering in the two counties (Richland and Lexington) that Columbia occupies.

Type	Buildings in Richland	in Lexington	Total	People sheltered Per building	Number
Permanent residents					516,251
Schools—general *	83	113	196	900	176,400
Hotels / motels	80	32	112	500	56,000
Churches	568	386	954	250	238,500
Schools—other**	63	16	79	900	71,100
				Total	1,058,251

*We assume that schools average 600 students and can shelter 900.

**Includes academies but excludes beauty schools, trade schools, driving schools, etc.

However, Charlotte NC had only a very small increase in population due to evacuation (from 396,000 to 411,000). The people that Columbia cannot shelter can easily find shelter in Charlotte.

Restricting Vehicle Types and Vehicle Numbers

Restrictions on numbers and types of vehicles would indeed increase the speed of the evacuation. However, there are no reliable ways to enforce such restrictions. Consider the following arguments:

- Forbidding camper vehicles may be unsuccessful, since for a sizable fraction of tourists the camper is their only vehicle.
- Restricting the number of vehicles to one per family:
  - The record-keeping involved would be prohibitive.
  - For some families, more than one vehicle is needed to carry all of the family members.

The I-95 Traffic Problem

We assume that if the interchange is not closed, at least 75% of the people coming up from Florida and Georgia on I-95 will take I-26 to Columbia. This is because the next major city reachable from I-95 is Raleigh, 150 mi further on. In our simulation, not closing this intersection (but keeping the eastbound lanes of I-26 reversed) increases the evacuation time to 19 h.

The Best Simulated Evacuation Plan

By altering various model parameters, we reduced the overall evacuation time to 13 h:

- Reverse the eastbound lanes of I-26.
- Close the exit on I-95 N leading to I-26 W.
- Limit the flow of traffic from Charleston to I-26 W.

The third item is necessary to reduce congestion along I-26 in the Charleston area. If too many cars are allowed on, the speed of traffic in Charleston drops significantly. Although this unlimited access results in a greater average speed on the section of I-26 between Charleston and the I-95 interchange, the slowdown in the Charleston area is exactly the type of backup that caused complaints in 1999 and resulted in a greater total time to evacuate the city.

Conclusions

It is possible to evacuate coastal South Carolina in 13 h. Assuming that a hurricane watch is issued 36 h prior to landfall, the state can allow an ample delay between voluntary evacuation announcement and a subsequent mandatory order. However, state agencies must take considerable action to ensure that the evacuation will go as planned:

- Close the interchange between I-26 and I-95. Traffic on I-26 must remain on I-26; traffic on I-95 must remain on I-95.
- The two eastbound lanes of I-26 must be reversed immediately upon the mandatory evacuation order.
- In Charleston, restrict entrance to I-26 to 15 cars / min at each entrance ramp.

Everyone in the areas to evacuate must be notified. Within South Carolina, the existing Emergency Alert System includes many radio stations that can inform the public of the incoming hurricane, the steps to take during evacuation, and which roads to use.

Residents must be more convinced to evacuate than they were during Hurricane Floyd. Appropriate measures must be taken to ensure that residents evacuate and evacuate far enough inland.

Model Strengths and Weaknesses

Strengths

The model's predictions have a number of features found in a real evacuation or other high-density traffic flow:

- An initial congested area around the entrance ramps gives way to a high-flow area when there is no entering traffic.
- Overall traffic speed in high-flow areas is around 35 mph.
- Merging traffic causes a major decrease in flow.

Weaknesses

The model does not take into account

- **city streets**, which are important in moving people from the highways to shelter in Columbia.
- **accidents**. A single accident or breakdown could result in several hours of delay. Tow trucks should be stationed at regular intervals along major roads.
- **local traffic on the non-controlled-access highways**, which would slow traffic on those roads.

References

Beltrami, Edward. 1998. *Mathematics for Dynamic Modeling.* 2nd ed. San Diego, CA: Academic.

City of Myrtle Beach: Frequently Asked Questions. 2000. `http://www.cityofmyrtlebeach.com/faq.html` . Accessed 12 February 2001.

Compton's Encyclopedia Online, vol. 3.0. South Carolina. `http://www.comptons.com/encyclopedia/ARTICLES/0150/01710336_A.html` . The Learning Company., Inc. Accessed 12 February 2001.

Cutter, Susan L., and Kirstin Dow. 2000. University of South Carolina: Quick Response Report 128: South Carolina's Response to Hurricane Floyd. `http://www.cla.sc.edu/geog/hrl/Quick%20Response%20Report.htm` . Accessed 12 February 2001.

________, Robert Oldendick, and Patrice Burns. 2000. University of South Carolina: Preliminary Report 1: South Carolina's Evacuation Experience with Hurricane Floyd. `http://www.cla.sc.edu/geog/hrl/Floyd_evacuation.html` . Accessed 10 February 2001.

Emergency Response Planning and Management, Inc. 2000. Assessment of South Carolina's experience with the 1999 hurricane season. `http://www.state.sc.us/epd/HurrBrief.pdf` . Accessed 9 February 2001.

Rand McNally Road Atlas: United States, Canada, Mexico. 1995. Skokie, IL: Rand McNally & Company.

`smartpages.com` .

South Carolina Asset Network. `http://www.scan21.com/hurricane_main.html` . Accessed 9 February 2001.

U.S. Army: Hurricane/Tropical Storm Classification Scale. `http://www.sam.usace.army.mil/op/opr/hurrclss.htm` . Accessed 10 February 2001.

U.S. Census Bureau: South Carolina Profiles (from the 1990 census). `http://www.census.gov/datamap/www/45.html` . Accessed 10 February 2001.

U.S. Census Bureau: Georgia Profiles (from the 1990 census). `http://www.census.gov/datamap/www/13.html` . Accessed 12 February 2001.

Students Develop Optimal Coastal Evacuation Plan

SOUTHFIELD, MICH., FEB. 12—During September 13–15, 1999, Hurricane Floyd threatened landfall along the coast of South Carolina. In response to weather advisories and a mandatory evacuation order from the governor, hundreds of thousands of people simultaneously attempted to evacuate the coastal regions including Charleston and Myrtle Beach, causing unprecedented traffic jams along major highways. Although the evacuation was successful in that no lives were lost (largely since Floyd did not have as great an impact in the expected area), the evacuation was a failure in that it was not executed quickly nor completely enough to ensure the safety and well-being of all evacuating citizens had Hurricane Floyd made landfall in the Charleston area.

Since that problematic evacuation in 1999, state officials have been working on plans for a safe, efficient evacuation of the South Carolina coast, preparing for the event that a hurricane like Floyd threatens the coast again. They posed the problem to teams of mathematicians all over the country.

After working for four days, a group of talented students evolved a specific plan to safely and quickly evacuate every coastal county in South Carolina (nearly 1 million people) within 13 hours, using a computer simulation of their own design. The plan involves the reversal of the two coastal-bound lanes on Interstate 26 (the main east-west highway), as well as traffic control and detours throughout the major roads heading inland.

The students' plan guides the mass traffic flow to areas the students felt were capable of sheltering large numbers of evacuees. The main destination was Columbia, the capital and largest inland city in South Carolina. Other destinations were Spartanburg, Florence, Sumter, and Greenville in South Carolina; Augusta in Georgia; and Charlotte in North Carolina. The plan also accounted for the possibility of very heavy traffic coming northward from Georgia and Florida on I-95, fleeing from the same hurricane, which could adversely affect the evacuation in South Carolina.

Additionally, the students set forth plans to shelter the more than 1 million people who would be in Columbia after the evacuation is complete. By making use of all the city's schools, hotels, motels, and churches as shelters, nearly all the evacuees could be sheltered. The few remaining evacuees could easily find shelter north in Charlotte, which in 1999 received few evacuees.

— Mark Wagner, Kenneth Kopp, and William E. Kolasa in Southfield, Mich.

2002: The Airline Overbooking Problem

You're all packed and ready to go on a trip to visit your best friend in New York City. After you check in at the ticket counter, the airline clerk announces that your flight has been overbooked. Passengers need to check in immediately to determine if they still have a seat.

Historically, airlines know that only a certain percentage of passengers who have made reservations on a particular flight will actually take that flight. Consequently, most airlines overbook—that is, they take more reservations than the capacity of the aircraft. Occasionally, more passengers will want to take a flight than the capacity of the plane, leading to one or more passengers being bumped and thus unable to take the flight for which they had reservations.

Airlines deal with bumped passengers in various ways. Some are given nothing, some are booked on later flights on other airlines, and some are given some kind of cash or airline ticket incentive.

Consider the overbooking issue in light of the current situation:

- fewer flights by airlines from point A to point B;
- heightened security at and around airports,
- passengers' fear, and
- loss of billions of dollars in revenue by airlines to date.

Build a mathematical model that examines the effects that different overbooking schemes have on the revenue received by an airline company, in order to find an optimal overbooking strategy—that is, the number of people by which an airline should overbook a particular flight so that the company's revenue is maximized. Ensure that your model reflects the issues above and consider alternatives for handling "bumped" passengers. Additionally, write a short memorandum to the airline's CEO summarizing your findings and analysis.

Comments

The Outstanding papers were by teams from Bethel College, Duke University, Harvey Mudd College, University of Colorado, and Wake Forest University (two teams). Their papers, together with an author-judge's commentary, were published in *The UMAP Journal* 23 (3) (2002): 273–370.

Problem Origin

The problem was contributed by William P. Fox (Francis Marion University).

Author-Judge's Comments

No paper analyzed every element nor applied critical validation and sensitivity analysis to all aspects of their model. Judges found many papers *with the exact same model (down to the exact same letters used for the variables) and none of these clearly cited the universal source anywhere in the submission*. The failure to properly credit the original source critically hurt these papers; it was obvious their basic model was not theirs but came from a published source.

The required elements, as viewed by the judges, were to

- develop a basic overbooking model that enabled one to find optimal values,
- consider alternative strategies for handling overbooked passengers,
- reflect on post-9-11 issues, and
- contain the CEO report of finding and analysis.

Most of the better papers did an extensive literature and Web search concerning overbooking by airlines and used this information in their model building.

> *The poorest section in all papers, including many of the Outstanding papers, was the section on assumptions with rational justification.*

Many papers just skipped this section and went directly from the problem to model-building!

Most papers used a stochastic approach for their model. With interarrival times assumed to be exponential, a Poisson process was often used to model passengers. Teams moved quickly from the Poisson to a binomial distribution with p and $1 - p$ representing "shows" and "no-shows" for ticket-holders. Many teams started directly with the binomial distribution without loss of continuity. Some teams went on to use the normal approximation to the binomial. Revenues were generally calculated using some sort of "expected value" equation. Some teams built nonlinear optimization models, which was a nice and different approach.

Teams usually started with a simple example: a single plane with a fixed cost and capacity, one ticket price, and a reasonable value for no-shows based on historical data. This then became a model from which teams could build refinements (not only to their parameters) but also to include the changes based on post-9-11.

Teams often simulated these results using the computer and then made sense of the simulation by summarizing the results.

Wake Forest had two Outstanding papers. Both papers began using a binomial approach as their base model. The paper entitled "ACE is High" was the INFORMS winner because of its superior analysis. The team did a superb job in maximizing revenue after examining alternatives and varying parameters. The other paper, entitled "Bumping for Dollars," developed a single-plane model, a two-plane model, and generalized to an n-plane model.

The Harvey Mudd team, the MAA winner, had—by far—the best literature search. They used it to discuss existing models to determine if any could be used for post-9-11. Their research examined many of the current overbooking models that could be adapted to the situation.

The University of Colorado team used the real Denver-based Frontier Airlines as their airline. They began with the binomial random variable approach, with revenues being expected values. They modeled both linear and nonlinear compensation plans for bumped passengers. They developed an auction-style model using Chebyshev's weighting distribution. They also consider time-dependency in their model.

The Duke University team, the SIAM winner, had an excellent mix of literature search material and development of their own models. They too began with a basic binomial model. They considered multiple fares and related each post-9-11 issue to parameters in their model. They varied their parameters and provided many key insights to the overbooking problem. This paper was the first paper in many years to receive an Outstanding rating from each judge who read the paper.

The Bethel College team built a risk assessment model. They used a normal distribution as their probability distribution and then put together an expected value model for revenue. Their analysis of Vanguard Airlines with a plane capacity of 130 passengers was done well.

Most papers found that an "optimal" overbooking strategy overbooks by between 9% and 15%, and they used these numbers to find "optimal" revenues. Many teams tried alternative strategies for compensation, and some even distinguished classes of seats on an airplane.

Models for Evaluating Airline Overbooking

Michael P. Schubmehl
Wesley M. Turner
Daniel M. Boylan
Harvey Mudd College
Claremont, CA

Advisor: Michael E. Moody

Introduction

We develop two models to evaluate overbooking policies.

The first model predicts the effectiveness of a proposed overbooking scheme, using the concept of expected marginal seat revenue (EMSR). This model solves the discount seat allocation problem in the presence of overbooking factors for each fare class and evaluates an overbooking policy stochastically.

The second model takes in historical flight data and reconstructs what the optimal seat allocation should have been. The percentage of overbooking revenue obtained in practice serves as a measure of the policy's value.

Finally, we examine the overbooking problem in light of the recent drastic changes to airline industry and conclude that increased overbooking would bring short-term profits to most carriers. However, the long-term ill effects that have traditionally caused airlines to shun such a policy would be even more pronounced in a post-tragedy climate.

Literature Review

There are two major ways that airlines try to maximize revenues: overbooking (selling more seats than available on a given flight) and seat allocation (price discrimination). These measures are believed to save major airlines as much as half a billion dollars each year, in an industry with a typical yearly profit of about $1 billion dollars [Belobaba 1989].

Beckman [1958] models booking and no-shows in an effort to find an optimal overbooking strategy. He ignores advance cancellations, assuming that all cancellations are no-shows [Rothstein 1985]. A method that is easy to implement but sophisticated enough to allow for cancellations and group reservations was developed by Taylor [1962]. Versions of this model were implemented at Iberia Airlines [Shlifer and Vardi 1975], British Overseas Airways Corporation, and El Al Airlines [Rothstein 1985].

None of these approaches considers multiple fare classes. Littlewood [1972] offers a simple two-fare allocation rule: A discount fare should be sold only if the discount fare is greater than or equal to the *expected marginal return* from selling the seat at full-fare. This idea was generalized by Belobaba [1989] to include any number of fare classes and allow the integration of overbooking. We use expected marginal seat revenue in predicting about overbooking schemes.

There is a multitude of work on the subject [McGill 1999]—according to Weatherford and Bodily [1992], there are more than 124,416 classes of models for variations of the yield management problem, though research has settled into just a few of these. Several authors tried to better Belobaba's [1987] heuristic in the presence of three or more fare classes (for which it is demonstrably suboptimal) [Weatherford and Bodily 1992]; generally, these adaptive methods for obtaining optimal booking limits for single-leg flights are done by dynamic programming [Mcgill 1999].

After deregulation in 1978, airlines were no longer required to maintain a direct-route system to major cities. Many shifted to a hub-and-spoke system, and network effects began to grow more important. To maximize revenue, an airline may want to consider a passenger's full itinerary before accepting or denying their ticket request for a particular leg. For example, an airline might prefer to book a discount fare rather than one at full price if the passenger is continuing on to another destination (and thus paying an additional fare).

The first implementations of the origin-destination control problem considered segments of flights. The advantage to this was that a segment could be blacked out to a particular fare class, lowering the overall complexity of a booking scheme. Another method, *virtual nesting*, combines fare classes and flight schedules into distinct buckets [Mcgill 1999]. Inventory control on these buckets would then give revenue-increasing results. Finally, the bid-price method deterministically assigns a value to different seats on a flight leg. The legs in an itinerary are then summed to establish a bid-price for that itinerary; a ticket request is accepted only if the fare exceeds the bid-price [Mcgill 1999]

The most realistic yield management problem takes into account five price classes. The ticket demands for different fare classes are randomized and correlated with one other to allow for sell-ups and the recapture of rejected customers on later flights. Passengers can no-show or cancel at any time. Group reservations are treated separately from individuals—their cancellation probability distribution is likely different. Currently, most work assumes that passengers who pay full fare would not first check for availability of a lower-class ticket; a more realistic model would allow buyers of a higher-class ticket to be diverted by a lower fare. A full accounting of network effects would consider the relative value of what Weatherford and Bodily [1992] terms *displacement*—denying a discount passenger's ticket request to fly a multileg itinerary in favor of leaving one of the legs open to a full-fare passenger.

Unfortunately, while the algorithms for allocating seats and setting overbooking levels are highly developed, there has been little work done on the problem of evaluating how effective these measures actually are. Our solution applies industry-standard methods to find optimal booking levels, then examines the actual booking requests for a given flight to determine how close to an optimal revenue level the scheme actually comes.

Factors Affecting Overbooking Policy

General Concerns

The reason that overbooking is so important is because of multiple fare classes. With only one fare class, it would be easier for airlines to penalize customers for no-shows. However, while most airlines offer nonrefundable discount tickets, they prefer not to penalize those who pay full fare, like business travelers, because these passengers account for most of the profits.

The overbooking level of a plane is dictated by the likelihood of cancellations and of no-shows. An overbooking model compares the revenue generated by accepting additional reservations with the costs associated with the risk of overselling and decides whether additional sales are advisable. In addition, the "recapture" possibility can be considered, which is the probability that a passenger denied a ticket will simply buy a ticket for one of the airline's other flights. Since a passenger is more valuable to the airline buying a ticket on a flight that has empty seats to fill than on one that is already overbooked, a high recapture probability reduces the optimal overbooking level [Smith et al. 1992].

No major airline overbooks at profit-maximizing levels, because it could not realistically handle the problems associated with all the overloaded flights. This gives the overbooking optimization problem some important constraints. The total flight revenue is to be maximized, subject to the conditions that only a certain portion of flights have even one passenger denied boarding (one oversale), and that a bound is placed on the expected total number of oversales. Dealing with even one oversale is a hassle for the airline's staff, and they are not equipped to handle such problems on a large scale. Additionally, some research indicates that increased overbooking levels would most likely trigger an "overbooking war" [Suzuki 2002], which would increase short-term profits but would probably engender enough consumer resentment that the industry as a whole would lose business.

While the overbooking problem sets a limit for sales on a flight as a whole, proper seat allocation sets an optimal point at which to stop selling tickets for individual fare levels. For example, a perfectly overbooked plane, loaded exactly to capacity, could be flying at far below its optimal revenue level if too many discount tickets were sold. The more expensive tickets are not for first-class seats and involve no additional luxuries above the discount tickets, apart from more lenient cancellation policies and the ability to buy the tickets a shorter time before the flight's departure.

September 11, 2001

Since the September 11 terrorist attacks, there have been significant changes in the airline business. In addition to the forced cancellation of many flights in the immediate aftermath of the attacks and the extreme levels of cancellations and no-shows by passengers after that, passenger traffic has dropped sharply in general. The huge downturn in passenger levels has led to large reductions in service by most carriers.

In terms of the booking problem, there are fewer flights for passengers to spill over onto, which could increase the loss by an airline if it bumps a passenger from a flight. On the other hand, since passenger levels have fallen so far, it is less likely that an airline will overfill any given flight. The heightened security at airports will likely increase the passenger no-show rate somewhat, as passengers get delayed at security checkpoints. At the very least, it should almost completely remove the problem of "go-shows," passengers who show up for a flight but are not in the airline's records.

On the whole, optimal booking strategies have become even more vital as airlines have already lost billions of dollars, and some teeter on the brink of failure. Some overbooking tactics previously dismissed as too harmful in the long run might be worthwhile for companies in trouble. For example, an airline near failure might increase the overbooking rate to the level that maximizes revenue, without regard to the inconvenience and possible future resentment of its customers.

Constructing the Model

Objectives

A scheme for evaluating overbooking policies needs to answer two questions: how well should a *new* overbooking method perform, and how well is a *current* overbooking scheme already working? The first is best addressed by a simple model of an airline's booking procedures; given some setup for allocating seats to fare classes, candidate overbooking schemes can be laid on top and tested by simulation. This approach has the advantage that it provides insight into *why* an overbooking scheme is or is not effective and helps to illuminate the characteristics of an optimal overbooking approach.

The second question is, in some respects, a simpler one to answer. Given the actual (over)booking limits that were imposed on each fare class, and all available information on the actual requests for reservations, how much revenue might have been gained from overbooking, compared to how much actually was? This provides a very tangible measure of overbooking performance but very little insight into the reasons for results.

The enormous number of factors affecting the design and evaluation of an overbooking policy force us to make simplifying assumptions to construct models that meet both of these goals.

Assumptions

- **Fleet-wide revenues can be near-optimized one leg at a time.**

  Maximizing revenue involves complicated interactions between flights. For instance, a passenger purchasing a cheap ticket on a flight into a major hub might actually be worth more to the airline than a business-class passenger, on account of connecting flights. We assume that such effects can be compensated for by placing passengers into fare classes based on revenue potential rather than on the fare for any given leg. This assumption effectively reduces the network problem to a single-leg optimization problem.

- **Airlines set fares optimally.**

  Revenue maximization depends strongly on the prices of various classes of tickets. To avoid getting into the economics of price competition and supply / demand, we assume that airlines set prices optimally. This reduces revenue maximization to setting optimal fare-class (over)booking limits.

- **Historical demand data are available and applicable.**

  The model needs to estimate future demand for tickets on any given flight. We assume that historical data are available on the number of tickets sold any given number of days t before a flight's departure. In some respects, this assumption is unrealistic because of the problem of data *censorship*—that is, the failure of airlines to record requests beyond the booking limit for a fare class [Belobaba 1989]. On the other hand, statistical methods can be used to reconstruct this information [Boeing Commercial Airline Company 1982, 7–16; Swan 1990].

- **Low-fare passengers tend to book before high-fare ones.**

  Discount tickets are often sold under advance purchase restrictions, for the precise reason that it enables price discrimination. Because of restrictions like these, and because travelers who plan ahead search for cheap tickets, low-fare passengers tend to book before high-fare ones.

Predicting Overbooking Effectiveness

Disentangling the effects of overbooking, seat allocation, pricing schemes, and external factors on revenues of an airline is extremely complicated. To isolate the effects of overbooking as much as possible, we want a simple, well-understood seat allocation model that provides an easy way to incorporate various overbooking schemes. In light of this objective, we pass up several methods for finding optimal booking limits on single-leg flights detailed in, for example, Curry [1990] and Brumelle [1993], in favor of the simpler expected marginal seat revenue (EMSR) method [Belobaba 1989].

EMSR was developed as an extension of the well-known rule of thumb, popularized by Littlewood [1972], that revenues are maximized in a two-fare system by capping sales of the lower-class ticket when the revenue from selling an additional lower-class ticket is balanced by the *expected* revenue from selling the same seat as an upper-class ticket. In the EMSR formulation, any number of fare classes are permitted and the goal is "to determine how many seats *not to sell* in the lowest fare classes and to retain for *possible* sale in higher fare classes closer to departure day" [Belobaba 1989].

The only information required to calculate booking levels in the EMSR model is a probability density function for the number of requests that will arrive before the flight departs, in each fare class and as a function of time. For simplicity, this distribution can be assumed to be normal, with a mean and standard deviation that change as a function of the time remaining. Thus, the only information an airline would need is a historical average and standard deviation of demand in each class as a function of time. Ideally, the information would reflect previous instances of the particular flight in question. Let the mean and standard deviations in question be denoted by $\mu_i(t)$ and $\sigma_i(t)$ for each fare class $i = 1, 2, \ldots, k$. Then the probability that demand is greater than some specified level S_i is given by

$$\bar{P}_i(S_i, t) \equiv \frac{1}{\sqrt{2\pi}\,\sigma_i(t)} \int_{S_i}^{\infty} e^{(r-\mu_i(t))^2/\sigma_i(t)^2}\, dr.$$

This *spill probability* is the likelihood that the S_ith ticket would be sold if offered in the ith category. If we further allow $f_i(t)$ to denote the expected revenue resulting from a sale to class i at a time t days prior to departure, we can define

$$\mathrm{EMSR}_i(S_i, t) = f_i(t) \cdot \bar{P}_i(S_i, t),$$

or simply the revenue for a ticket in class i times the probability that the S_ith seat will be sold. The problem, however, is to find the number of tickets S_j^i that should be protected from the lower class j for sale to the upper class i (ignoring other classes for the moment). The optimal value for S_j^i satisfies

$$\mathrm{EMSR}_i(S_j^i, t) = f_j(t), \qquad \textbf{(1)}$$

so that the expected marginal revenue from holding the S_j^ith seat for class i is exactly equal to (in practice, slightly greater than) the revenue from selling it immediately to someone in the lower class j. The booking limits that should be enforced can be derived easily from the optimal S_j^i values by letting the booking limit B_j for class j be

$$B_j(t) = C - S_j^{j+1} - \sum_{i<j} b_i(t), \qquad \textbf{(2)}$$

that is, the capacity C of the plane, less the protection level of the class above j from class j and less the total number of seats already reserved. Sample EMSR curves, with booking limits calculated in this fashion, are shown in **Figure 1**.

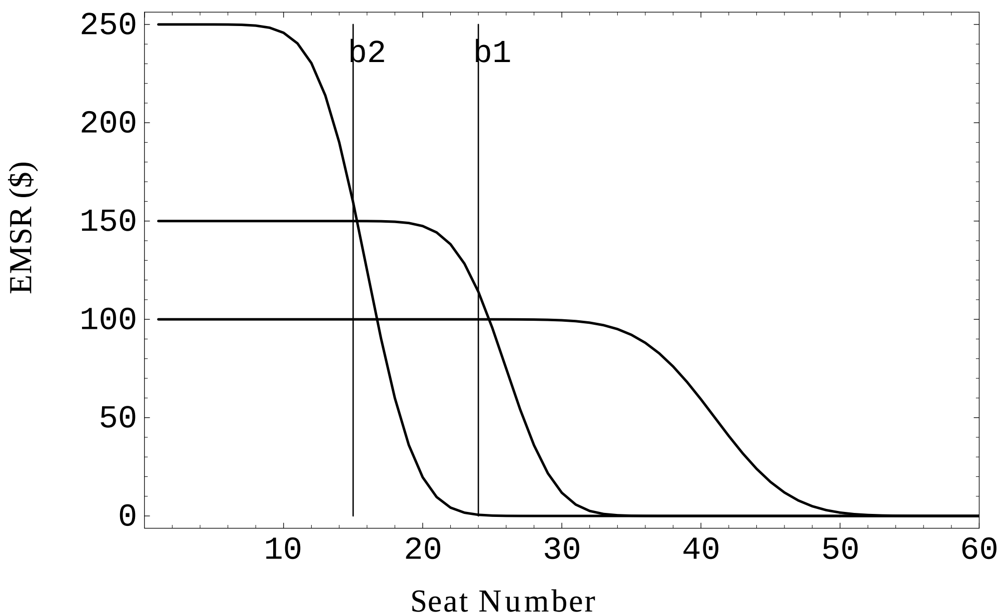


Figure 1. Expected marginal seat revenue (EMSR) curves for three class levels, with the highest-revenue class at the top. Each curve represents the revenue expected from protecting a particular seat to sell to that class. Also shown are the resulting booking limits for each of the lower classes—that is, the levels at which sales to the lower class should stop to save seats for higher fares.

This formulation does not account for overbooking; if we allow each fare class i to be overbooked by some factor OV_i, the optimality condition **(1)** becomes

$$\mathrm{EMSR}_i(S_j^i, t) = f_j(t) \cdot \frac{OV_i}{OV_j}. \tag{3}$$

This can be understood in terms of an adjustment to f_i and f_j; the overbooking factors are essentially cancellation probabilities, so we use each OV_i to deflate the expected revenue from fare class i. Then

$$\bar{P}_i(S_j^i, t) \cdot \frac{f_i(t)}{OV_i} = f_j(t) \cdot \frac{f_j(t)}{OV_j},$$

which is equivalent to **(3)**. Note that the use of a single overbooking factor for the entire cabin (that is, $OV_i = OV$) causes the OV_i and OV_j in **(3)** to cancel. Nonetheless, the boarding limits for each class are affected, because the capacity of the plane C must be adjusted to account for the extra reservations, so now

$$C^* = C \cdot OV$$

and the booking limits in **(2)** are adjusted upward by replacing C with C^*.

The EMSR formalism gives us the power to evaluate an overbooking scheme theoretically by plugging its recommendations into a well-understood stable model and evaluating them. Given the EMSR boarding limits, which can be updated dynamically as booking progresses, and the prescribed overbooking factors, a simulated string of requests can be handled. Since the EMSR model involves only periodic updates to establish limits that are fixed over the course of a day or so, a set of n requests can be handled with two lookups each (booking limit and current booking level), one subtraction, and one comparison; so all n requests can be processed on $\mathcal{O}(n)$ time. An EMSR-based approach would thus be practical in a real-world real-time airline reservations system, which often handles as many as 5,000 requests per second. Indeed, systems derived from EMSR have been adopted by many airlines [Mcgill 1999].

Evaluating Past Overbookings

The problem of evaluating an overbooking scheme that has already been implemented is somewhat less well studied than the problem of theoretically evaluating an overbooking policy. One simple approach, developed by American Airlines in 1992, measures the optimality of overbooking and seat allocation separately [Smith et al. 1992]. Their overbooking evaluation process assumes optimal seat allocation and, conversely, their seat allocation evaluation scheme assumes optimal overbooking. Under this assumption, an overbooking scheme is evaluated by estimating the revenue under *optimal* overbooking in two ways:

- If a flight is fully loaded and no passenger is denied boarding, the flight is considered to be optimally overbooked and to have achieved maximum revenue.
- If n passengers are denied boarding, the money lost due to bumping these passengers is added back in and the n lowest fares paid by passengers for the flight are subtracted from revenue.
- On the other hand, if there are n empty seats on the plane, the n highest-fare tickets that were requested but not sold are added to create the maximum revenue figure.

Their seat-allocation model estimates the demand for each flight by calculating a theoretical demand for each fare class and then setting the minimum flight revenue (by filling the seats lowest-class first) and the maximum flight revenue (by filling the seats highest-class first). To estimate demand, we use the information on the flight's sales up to the point where each class closed. By assuming that demand is increasing for each class, we can project the number of requests that would have occurred had the booking limits been disregarded.

Given these projected additional requests and the actual requests received before closing, it is straightforward to compute the best- and worst-case overbooking scenarios. The worst-case revenue R_- is determined by using no booking limits and taking reservations as they come, and the best-case revenue R_+ is determined by accommodating high-fare passengers first, giving the leftovers to the lower classes. The difference between these two figures is the revenue to be gained by the use of booking limits. Thus, the performance of a booking scheme that generates revenue R is

$$p = \frac{R - R_-}{R_+ - R_-} \cdot 100\%, \tag{4}$$

representing the percentage of the possible booking revenue actually achieved.

We select this method for evaluating booking schemes after the fact.

Analysis of the Models

Tests and Simulations

The EMSR method requires information on demand as a function of time. Although readily available to an airline, it is not widely published in a detailed form. Li [2001] provides enough data to construct a rough piecewise-linear picture of demand remaining as a function of time, shown in **Figure 2**.

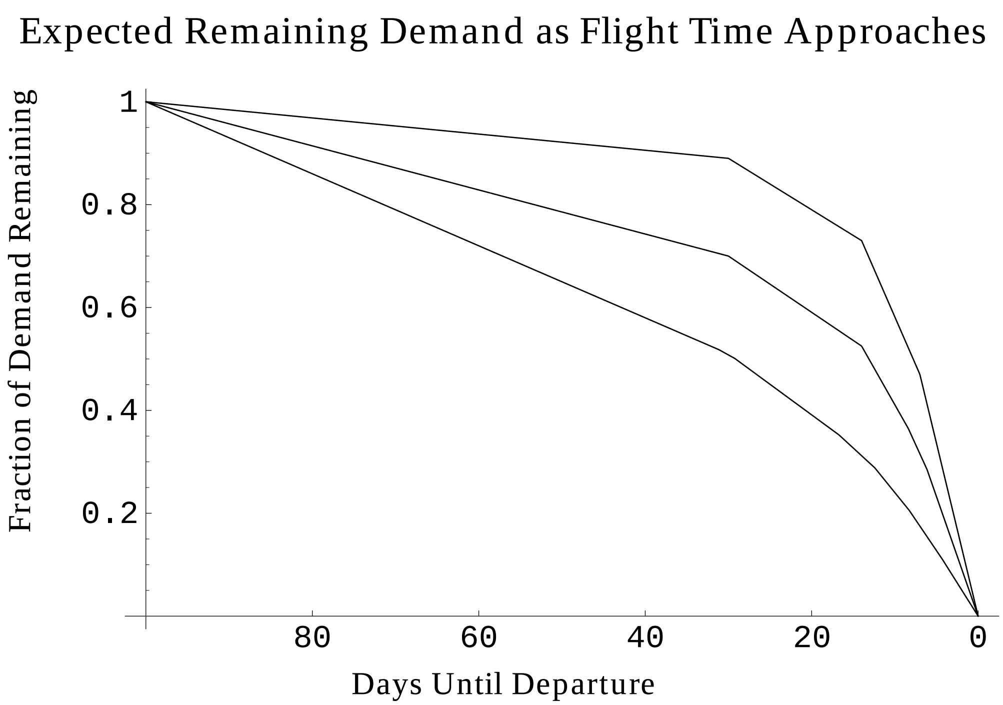


Figure 2. Demand remaining as a function of time for each of three fare classes, with the highest fare class on top. The curves represent the fraction of tickets that have yet to be purchased. Note that, for example, demand for high fare tickets kicks in much later than low-fare demand. (Data interpolated from Li [2001].)

This information can be inputted into the EMSR model to produce optimal booking limits that evolve in time. A typical situation near the beginning of ticket sales was shown in **Figure 1**, while the evolution of the limits themselves is plotted in **Figure 3**.

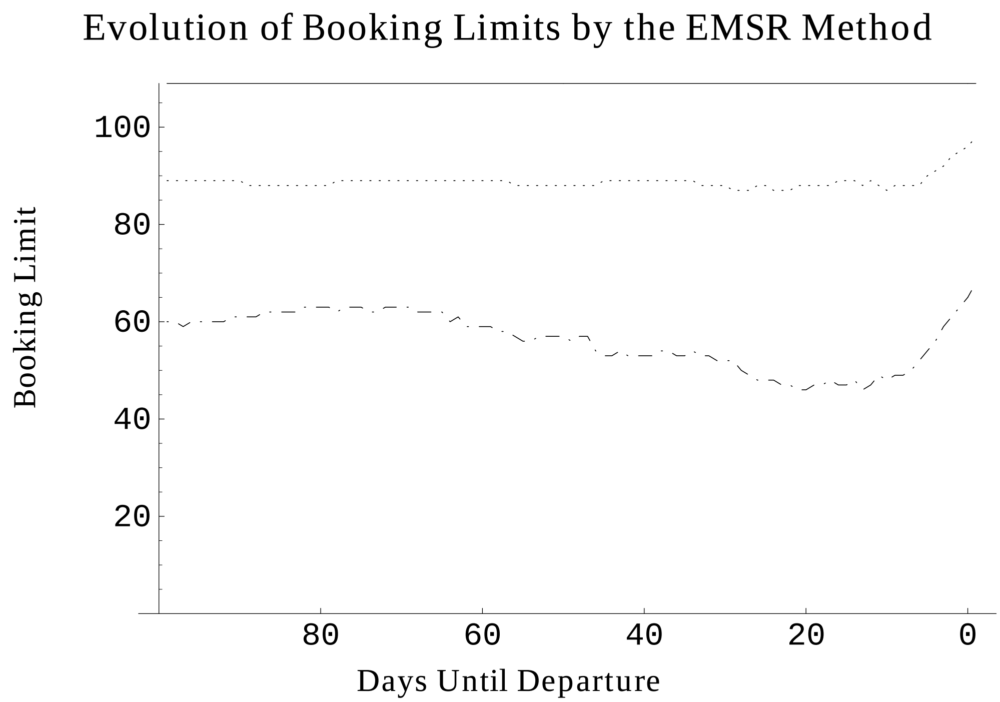


Figure 3. Booking limits for each class are dynamically adjusted to account for tickets already sold. For illustrative purposes, the number of tickets already sold is replaced here with the number of tickets that should have been sold according to expectations. In this case, the booking limits estimated at the beginning of the process are fairly accurate and require relatively little updating.

The demand information in **Figure 2** can also be used to simulate requests for reservations. By taking the difference between the demand remaining at day t and at day $(t-1)$ before departure, the expected demand on day t can be determined. The actual number of requests generated on that day is then given by a Poisson random variable with parameter λ equal to the expected number of sales [Rothstein 1971]. The requests generated in this manner can be passed to a request-handling simulation that looks at the most current booking limits and then accepts or denies ticket requests based on the number of reservations already confirmed and the reservations limit. An example of this booking process is illustrated in **Figure 4**.

The results of the booking process provide an ideal testbed for the revenue opportunity model employed to evaluate overbooking performance. The simulation conducted for **Figure 4** had demand values of $\{11, 41, 57\}$, for classes 1, 2, and 3 respectively, before ticket sales were capped. A linear forward projection of these sales rates indicates that they would have reached $\{18, 49, 69\}$ had the classes remained open. Given fare classes $\{\$250, \$150, \$100\}$, the minimum

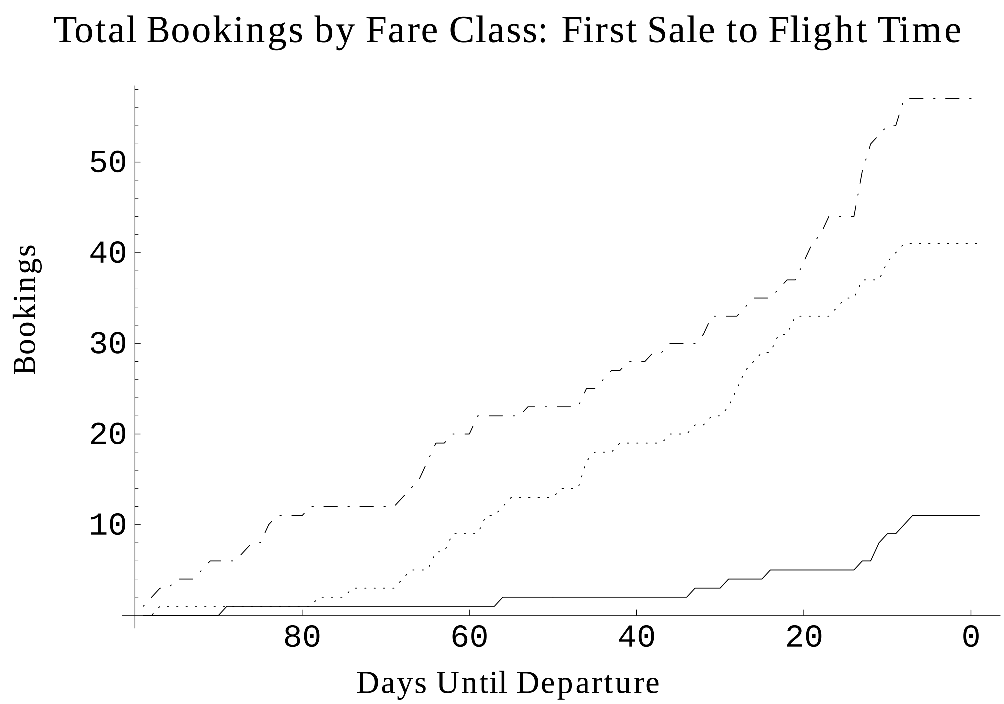


Figure 4. The EMSR-based booking limits are used to decide whether to accept or reject a sequence of ticket requests. These requests follow a Poisson distribution where the parameter λ varies with time to match the expected demand. Each fare class reaches its booking limit, as desired, so the flight is exactly full. Incorporating overbooking factors shifts the limits up accordingly.

revenue would be

$$R_- = \$100(69) + \$150(40) + \$250(0) = \$12,900$$

and the maximum revenue would be

$$R_+ = \$250(18) + \$150(49) + \$100(42) = \$16,050.$$

The actual revenue according to the EMSR formalism was

$$R = \$100(57) + \$150(41) + \$250(11) = \$14,600,$$

so the efficiency is

$$p = \frac{R - R_-}{R_+ - R_-} \cdot 100\% = \frac{\$14,600 - \$12,900}{\$16,050 - \$12,900} \cdot 100\% = 54\%,$$

without the use of a complicated overbooking scheme. This is not close to the efficiencies reported in Smith et al. [1992], which cluster around 92%. This relative inefficiency is to be expected, however, from a simplified booking scheme given incomplete booking request data.

Strengths and Weaknesses

Strengths

- **Applies widely accepted, industry-standard techniques.**

  Although more advanced (and optimal) algorithms are available and are used, EMSR and its descendants are still widely used in industry and can come close to optimality. The EMSR scheme, tested as-is on a real airline, caused revenue gains as much as 15% [Belobaba 1989].

  Our method for determining the optimality of a scheme after the fact is also based on tried and true methods developed by American Airlines [Smith et al. 1992].

- **Simplicity**

  Since it does not take into account as many factors as other booking models, EMSR is easier to deal with computationally. While a simple model may not be able to model a major airline with complete accuracy, an optimal pricing scheme can be made using only three fare classes [Li 2001].

Weaknesses

- **Neglects network effects**

  We treat the problem of optimizing each flight as if it were an independent problem although it is not.

- **Ignores sell-ups**

  In considering the discount seat allocation problem, we treat the demands for the fare classes as constants, independent of one other. This is not the case, because of the possibility of sell-ups. If the number of tickets sold in a lower fare class is restricted, then there is some probability that a customer requesting a ticket in that class will buy a ticket at a more expensive fare. This means it is possible to convert low-fare demand into high-fare demand, which would suggest protecting a higher number of seats for high fares than calculated by the model that we use. Sell-ups would be straightforward to incorporate into EMSR, but doing so would require additional information [Belobaba 1989].

- **Discounts possibility of recapture**

  Similar to sell-ups, the recapture probability is the probability that a passenger unable to buy a ticket at a certain price on a given flight will buy a ticket on a different flight. Depending on the recapture probability for each fare class, more or fewer seats might be allocated to discount fares.

Recommendations on Bumping Policy

In 1999, an average of only 0.88 passengers per 10,000 boardings were involuntarily bumped. Airlines are not required to keep records of the number of voluntary bumps, so it is impossible to determine a general bump rate.

Before bumping passengers involuntarily, the airline is required to ask for volunteers. Because there are no regulations on compensation for voluntary bumps, this is often a cheaper and more attractive method for airlines anyway. If too few people volunteer, the airline must pay those denied boarding 200% of the sum of the values of the passengers' remaining flight coupons, with a maximum of $400. This maximum is decreased to $200 if the airline arranges for a flight that will arrive less than 2 hours after the original flight. The airline may also substitute the offer of free or reduced fare transportation in the future, provided that the value of the offer is greater than the cash payment otherwise required. Alternatively, the airline may simply arrange alternative transportation if it is scheduled to arrive less than an hour after the original flight.

Auctions in which the airline offers progressively higher compensation for passengers who give up their seats are both the cheapest and the most common practice. As long as the airline does not engage in so much overbooking that it cannot find suitable reroutes for passengers bumped from their original itineraries, no alternatives to this policy need to be considered.

Conclusions

The two models presented in this paper work together to evaluate overbooking schemes by simulating their effects in advance and by quantifying their effects after implementation.

The expected marginal seat revenue (EMSR) model predicts overbooking scheme effectiveness. It determines the correct levels of protection for each fare class above the lowest—that is, how many seats should be reserved for possible sale at later dates and higher fares. Overbooking factors can be specified separately for each fare class, so the model effectively takes in overbooking factors and produces booking limits that can be used to handle ticket requests.

The revenue opportunity model attempts to estimate the maximum revenue from a flight under perfect overbooking and discount allocation. This is accomplished by estimating the actual demand for seats, then calculating the revenue that these seats would generate if sold to the highest-paying customers. Simple calculations produce the ideal overbooking cap and the optimal discount allocation for the flight. Thus, this model effectively represents how the airline would sell tickets if they had perfect advance knowledge of demand.

After the September terrorist attacks and their subsequent catastrophic effects on the airline industry, heightened airport security and fearful passengers will increase no-show and cancellation rates, seeming to dictate increasing overbooking levels to reclaim lost profits.

Airlines considering such action should be cautioned, however, that the negative effects of increased overbooking could outweigh the benefits. With reduced airline service, finding alternative transportation for displaced passengers could be more difficult. The effect of denying boarding to more passengers, along with the greater inconvenience of being bumped, could seriously shake consumers' already-diminished faith in the airline industry. With airlines already losing huge numbers of customers, it would be a mistake to risk permanently losing them to alternatives like rail and auto travel by alienating them with frequent overselling.

References

Beckman, J.M. 1958. Decision and team problems in airline reservations. *Econometrica* 26: 134–145.

Belobaba, Peter. 1987. Airline yield management: An overview of seat inventory control. *Transportation Science* 21: 63–73.

________. 1989. Application of a probabilistic decision model to airline seat inventory control. *Operations Research* 37: 183–197.

Biyalogorsky, Eyal, et al. 1999. Overselling with opportunistic cancellations. *Marketing Science* 18 (4): 605–610.

Boeing Commercial Airline Company. 1982. Boeing Promotional Fare Management System: Analysis and Research Findings. Seattle, WA: Boeing, Seattle.

Bowen, B.D., and D.E. Headley. 2000. Airline Quality Rating 2000. `http://www.uomaha.edu/~unoai/research/aqr00/` .

Brumelle, S.L., and J.I. McGill. 1993. Airline seat allocation with multiple nested fare classes. *Operations Research* 41: 127–137.

Chatwin, Richard E. 2000. Optional dynamic pricing of perishable products with stochastic demand and a finite set of prices. *European Journal of Operations Research* 125: 149–174.

Curry, R.E. 1990. Optimal airline seat allocation with fare classes nested by origins and destinations. *Transportation Science* 24: 193–204).

Davis, Paul. Airline ties profitability yield to management. 1994. *Siam News* 27 (5); `http://www.siam.org/siamnews/mtc/mtc694.htm` .

Li, M.Z.F. 2001. Pricing non-storable perishable goods by using a purchase restriction with an application to airline fare pricing. *European Journal of Operations Research* 134: 631–647.

Littlewood, K. 1972. Forecasting and control of passenger bookings. *AGIFORS Symposium Proceedings*: 12.

McGill, J.I., and G.Z. and Van Ryzin. 1999. Revenue management: Research overview and prospects. *Transportation Science* 33 (2): 233–256.

Rothstein, M. 1971. An airline overbooking model. *Transportation Science* 5: 180–192.

Rothstein, Marvin 1985. OR and the airline overbooking problem. *Operations Research* 33 (2): 237–247.

Ross, Robert. 1998. Embry-Riddle experts reveal airfare secrets. `http://comm.db.erau.edu/leader/fall98/priced.html`.

Shlifer, R., and Y. Vardi. 1975. An Airline Overbooking Policy. *Transportation Science* 9: 101-114.

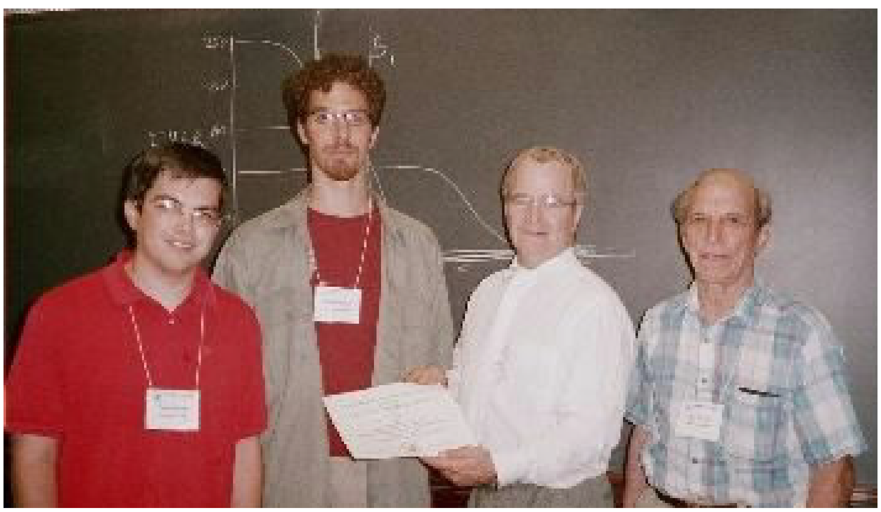
Swan. 1990. Revenue management forecasting biases. Working paper. Seattle, WA: Boeing Commercial Aircraft.

Smith, Barry C., et al. 1992. Yield management at American Airlines. *Interfaces* 22 (1): 8–31.

Suzuki, Yoshinori. 2002. An empirical analysis of the optimal overbooking policies for U.S. major airlines. *Transportation Research* E38: 135–149.

Taylor, C.J. 1962. The determination of passenger booking levels. *AGIFORS Symposium Proceedings*, vol. 2. Fregene, Italy.

Weatherford, L.R., and S.E. Bodily. 1992. A taxonomy and research overview of perishable-asset revenue management: Yield management, overbooking, and pricing. *Operations Research* 40: 831–844.



Presentation by Richard Neal (MAA Student Activities Committee Chair) of the MAA award to Daniel Boylan and Wesley Turner of the Harvey Mudd College team (Michael Schubmehl could not come), after their presentation at the MAA Mathfest in Burlington, VT, in August. On the right is Ben Fusaro, Founding Director of the MCM. Photo by Ruth Favro, Lawrence Tech University.

Letter to the CEO of a Major Airline

Airline overbooking is just one facet of a revenue management problem that has been studied extensively in operations research literature. Airlines have been practicing overbooking since the 1940's, but early models of overbooking considered only the most rudimentary cases. Most importantly, they did not take into account the revenue maximizing potential of price discrimination—charging different fares for identical seats. In order to maximize yield, it is particularly critical to price discriminate between business and leisure travelers. That is, when filling the plane, book as many full fare passengers and as few discount fare passengers as possible.

The implementation of a method of yield management can have dramatic effects on an airline's revenue. American Airlines managed its seat inventory to a $1.4 billion increase in revenue from 1989 to 1992—about 50% more than its net profit for the same period. Controlling the mix of fare products can translate into revenue increases of $200 million to $500 million for carriers with total revenues of $1 billion to $5 billion.

Though several decision models of airline booking have been developed over the years, comparing one scheme to another remains a difficult task. We have taken a two-pronged approach to this problem, both simulating and measuring a booking scheme's profitability.

In order to simulate a booking scheme's effect, we used the expected marginal seat revenue (EMSR) model proposed by Belobaba [1989] to generate near-optimal decisions on whether to accept or deny a ticket request in a given fare class. The EMSR model accepts as input overbooking levels for each of the fare classes that compose a flight, so different policies can be plugged in for testing.

Our approach to measuring a current scheme's profitability is similar to one used at American Airlines [Smith et al. 1992]. We compare the actual revenue generated by a flight with an ideal level calculated with the benefit of hindsight, as well as with a baseline level that would have been generated had no yield management been used. By calculating the percentage of this spread earned by a flight employing a particular scheme, we are able to gauge the effectiveness of different booking schemes.

It is our hope that these models will prove useful in evaluating your airline's overbooking policies. Simulations should provide insight into the properties of an effective scheme, and measurements after the fact will help to provide performance benchmarks. Finally, while it may be tempting to increase overbooking levels in order to compensate for lost revenues in the post-tragedy climate, our results indicate this will probably hurt long-term profits more than they will help.

Cordially,

Michael P. Schubmehl, Wesley M. Turner, and Daniel M. Boylan

2003: The Gamma Knife Treatment Problem

Planning

Stereotactic radiosurgery delivers a single high dose of ionizing radiation to a radiographically well-defined, small intracranial 3D brain tumor without delivering any significant fraction of the prescribed dose to the surrounding brain tissue. Three modalities are commonly used in this area; they are the gamma knife unit, heavy charged particle beams, and external high-energy photon beams from linear accelerators.

The gamma knife unit delivers a single high dose of ionizing radiation emanating from 201 cobalt-60 unit sources through a heavy helmet. All 201 beams simultaneously intersect at the isocenter, resulting in a spherical (approximately) dose distribution at the effective dose levels. Irradiating the isocenter to deliver dose is termed a"shot." Shots can be represented as different spheres. Four interchangeable outer collimator helmets with beam-channel diameters of 4, 8, 14, and 18 mm are available for irradiating different size volumes. For a target volume larger than one shot, multiple shots can be used to cover the entire target. In practice, most target volumes are treated with 1 to 15 shots. The target volume is a bounded, three-dimensional digital image that usually consists of millions of points.

The goal of radiosurgery is to deplete tumor cells while preserving normal structures. Since there are physical limitations and biological uncertainties involved in this therapy process, a treatment plan needs to account for all those limitations and uncertainties. In general, an optimal treatment plan is designed to meet the following requirements.

1. Minimize the dose gradient across the target volume.
2. Match specified isodose contours to the target volumes.
3. Match specified dose-volume constraints of the target and critical organ.
4. Minimize the integral dose to the entire volume of normal tissues or organs.
5. Constrain dose to specified normal tissue points below tolerance doses.
6. Minimize the maximum dose to critical volumes.

In gamma unit treatment planning, we have the following constraints:

1. Prohibit shots from protruding outside the target.
2. Prohibit shots from overlapping (to avoid hot spots).

3. Cover the target volume with effective dosage as much as possible. But at least 90% of the target volume must be covered by shots.

4. Use as few shots as possible.

Your tasks are to formulate the optimal treatment planning for a gamma knife unit as a sphere-packing problem, and propose an algorithm to find a solution. While designing your algorithm, you must keep in mind that your algorithm must be reasonably efficient.

Comments

The Outstanding papers were by teams from Donghua University (Shanghai, China), Peking University (Beijing, China), University of Colorado, University of Washington, and Youngstown State University. Their papers were published in *The UMAP Journal* 24 (3) (2003): 319–390.

Problem Origin

The problem was contributed by Jie Wang.

The Genetic Algorithm-Based Optimization Approach for Gamma Unit Treatment

Sun Fei
Yang Lin
Wang Hong
Donghua University
Shanghai, China

Advisor: Ding Yongsheng

Abstract

The gamma knife is used to treat brain tumors with radiation. The treatment planning process determines where to center the shots, how long to expose them, and what size focusing helmets should be used, to cover the target with sufficient dosage without overdosing normal tissue or surrounding sensitive structures.

We formulate the optimal treatment planning for a gamma-knife unit as a sphere-packing problem and propose a new genetic algorithm (GA)-based optimization approach for it. Considering the physical limitations and biological uncertainties involved, we outline a reasonable, efficient and robust solution.

First, we design a geometry-based heuristic to produce quickly a reasonable configuration of shot sizes, locations, and number. We first generate the skeleton using a 3D-skeleton algorithm. Then, along the skeleton, we use the GA-based shot placement algorithm to find a best location to place a shot. By continuously iterating the algorithm, we obtain the number, sizes, and the locations of all shots. After that, we develop a dose-based optimization method.

Then we implement simulations of our models in Matlab. We did numerous computer simulations, using different shape or size targets, to examine the effectiveness of our model. From the simulation results, we know that the geometry-based heuristic with the GA optimization approach is a useful tool for in the selection of the appropriate number of shots and helmet sizes. Generally, all of the optimized plans for various targets provide full-target coverage with 90% of the prescription isodose.

Moreover, we do sensitivity analysis to our model in the following aspects:

- the sensitive structures;
- at the 30%, 40%, 60%, or 70% isodose level;
- the issue of global versus local optimality;
- conformality; and
- the robustness. We also discuss the strengths and limitations of our model.

The results indicate that our approach is sufficiently robust and effective to be used in practice. In future work, we would fit ellipsoids instead of spheres, since some researchers note that the dose is skewed in certain directions.

Introduction

The gamma knife unit delivers ionizing radiation from 201 cobalt-60 sources through a heavy helmet. All beams simultaneously intersect at the *isocenter*, resulting in an approximately spherical dose distribution at the effective dose levels. Delivering dose is termed a *shot*, and a shot can be represented as a spheres. Four interchangeable outer collimator helmets with beam channel diameters of 4, 8, 14, and 18 mm are available. For a target volume larger than one shot, multiple shots can be used.

Gamma knife treatment plans are conventionally produced using a manual iterative approach. In each iteration, the planner attempts to determine

- the number of shots,
- the shot sizes,
- the shot locations, and
- the shot exposure times (weights) that would adequately cover the target and spare critical structures.

For large or irregularly shaped treatment volumes, this process is tedious and time-consuming, and the quality of the plan produced often depends on both the patience and the experience of the user. Consequently, a number of researchers have studied techniques for automating the gamma knife treatment planning process [Wu and Bourland 2000a; Shu et al. 1998]. The algorithms that have been tested include simulated annealing [Leichtman et al. 2000; Zhang et al. 2001], mixed integer programming, and nonlinear programming [Ferris et al. 2002; 2003; Shepard et al. 2000; Ferris and Shepard 2000].

The objective is to deliver a homogeneous (uniform) dose of radiation to the tumor (the target) area while avoiding unnecessary damage to the surrounding tissue and organs. Approximating each shot as a sphere [Cho et al. 1998] reduces the problem to one of geometric coverage. Kike [Liu and Tang 1997], we formulate optimal treatment planning as a sphere-packing problem and we propose an algorithm to determine shot locations and sizes.

Assumptions

To account for all physical limitations and biological uncertainties involved in the gamma knife therapy process, we make several assumptions as follows:

A1: The shape of the target is not too irregular, and the target volume is a bounded. As a rule of thumb, the target to be treated should be less than 35 mm in all dimensions. Its three-dimensional (3D) digital image, usually consisting of millions of points, can be obtained from a CT or MRI.

A2: We consider the target volume as a 3D grid of points and divide this grid into two subsets, the subset of points in and out of the target, denoted by T and N, respectively.

A3: Four interchangeable outer collimator helmets with beam channel diameters $w = \{4, 8, 14, 18\}$ mm are available for irradiating different size volumes. We use (x_s, y_s, z_s) to denote the coordinates of the center location of the shot and $t_{s,w}$ to denote the time (weight) that each shot is exposed. The total dose delivered is a linear function of $t_{s,w}$. For a target volume larger than one shot, multiple shots can be used to cover the entire target. There is a bound n on the number of shots, with typically $n \leq 15$.

A4: Neurosurgeons commonly use isodose curves as a means of judging the quality of a treatment plan; they may require that the entire target is surrounded by an isodose line of $x\%$, e.g., 30–70%. We use an isodose line of 50%, which means that the 50% line must surround the target.

A5: The dose cloud is approximated as a spherically symmetric distribution by averaging the profiles along x, y, and z axes. Other effects are ignored.

A6 The total dose deposited in the target and critical organ should be more than a fraction P of the total dose delivered; typically, $25\% \leq P \leq 40\%$.

Optimization Models

Analysis of the Problem

The goal of radiosurgery is to deplete tumor cells while preserving normal structures. An optimal treatment plan is designed to:

R1: match specified isodose contours to the target volumes;

R2: match specified dose-volume constraints of the target and critical organ;

R3: constrain dose to specified normal tissue points below tolerance doses;

R4: minimize the integral dose to the entire volume of normal tissues or organs;

R5: minimize the dose gradient across the target volume; and

R6: minimize the maximum dose to critical volumes.

It also is constrained to

C1: prohibit shots from protruding outside the target,

C2: prohibit shots from overlapping (to avoid hot spots),

C3: cover the target volume with effective dosage as much as possible (at least 90% of the target volume must be covered by shots), and

C4: use as few shots as possible.

We design the optimal treatment plan in two steps.

- We use a geometry-based heuristic to produce a reasonable configuration of shot number, sizes and locations.
- We use a dose-based optimization to produce the final treatment plan.

Geometry-Based Heuristic for Sphere-Packing

We model each shot as a sphere, and we use the medial axis transform (known as the *skeleton*) of the target volume to guide placement of the shots. The skeleton is frequently used in shape analysis and other related areas [Wu et al. 1996; Wu and Bourland 2000b; Zhou et al. 1998]. We use the skeleton just to find good locations of the shots quickly. The heuristic is in three stages:

- We generate the skeleton using a 3D skeleton algorithm.
- We place shots and choose their sizes along the skeleton to maximize a measure of our objective; this process is done by a genetic algorithm (GA)-based shot placement approach.
- After the number of focusing helmets to be included in the treatment plan is decided, the planning produces a list of the possible helmet combinations and a suggested number of shots to use.

Skeleton Generation

We adopt a 3D skeleton algorithm that follows similar procedures to Ferris et al. [2002]. We use a morphologic thinning approach [Wu 2000] to create the skeleton, as opposed to the Euclidean-distance technique. The first step in the skeleton generation is to compute the contour map containing distance information from the point to a nearest target boundary. Then, based on the contour map, several known skeleton extraction methods [Ferris et al. 2002; Wu et al. 1996; Wu and Bourland 2000b; Zhou et al. 1998; Wu 2000] can be used. Since the method in Ferris et al. [2002] is simple and fast, we use it.

Genetic Algorithm-Based Shot Placement

We restrict our attention to points on the skeleton. We start from a special type of skeleton point, an *endpoint* (**Figure 1**): A point in the skeleton is an endpoint if it has only one neighbor in the skeleton.

Starting from an endpoint, we look for the best point to place a shot and determine the shot size by using GA [Goldberg 1989; Mann et al. 1997]. In the GA-based shot placement algorithm, we must solve the following problems:

(a) an end point (b) an end point

Figure 1. Examples of endpoints.

The encoding method. In general, bit-string (0s and 1s)) encoding is the most common method adopted by GA researchers because of its simplicity and tractability. However, in this case, if we directly encode the point coordinates (x_s, y_s, z_s) into a bitstring, crossover and mutation generate some points that are not in the skeleton. To solve this problem, we build a table of correspondence between the point coordinates and the point number (1 to M); instead of encoding the point, we encode the point number. We select m points from all points of the skeleton to form a population; a single point is a *chromosome*.

Performance evaluation. The key to the GA-based approach is the fitness function. Ideally, we would like to place shots that cover the entire region without overdosing within (or outside) of the target. Overdosing occurs outside the target if we choose a shot size that is too large for the current location, and hence the shot protrudes from the target. Overdosing occurs within the target if we place two shots too close together for their chosen sizes.

Before defining a fitness function, we give some definitions:

- *Fraction*: A target part that is not large enough to be destroyed by the smallest shot without any harm to the surrounding normal tissue.
- *Span*: The minimum distance between the current location and the end-point at which we started.
- *Radius*: The approximate Euclidean distance to the target boundary.

We would like to ensure that the span, the radius, and the shot size w are as close as possible. Therefore, we choose a fitness function that is the sum of the squared differences between these three quantities. The fitness function can ensure that the generating fraction is the smallest after every shot is placed on the target [Ferris et al. 2002]:

$$\text{Fit} = \phi_{s,r}(x,y,z) + \phi_{s,w}(x,y,z) + \phi_{r,w}(x,y,z),$$

where

$$\phi_{s,r}(x,y,z) = \left[\text{span}(x,y,z) - \text{radius}(x,y,z)\right]^2, \quad \textbf{(1)}$$

$$\phi_{s,w}(x,y,z) = \left[\text{span}(x,y,z) - w\right]^2, \quad \textbf{(2)}$$

$$\phi_{r,w}(x,y,z) = \left[\text{radius}(x,y,z) - w\right]^2. \quad \textbf{(3)}$$

- Equation **(1)** ensures that we pack the target volume as well as possible, that is, the current span between shots should be close to the distance to the closest target boundary.
- Equation **(2)** is used to choose a helmet size that fits the skeleton best for the current location.
- Equation **(3)** favors a location that is the appropriate distance from the target boundary for the current shot size.

Genetic operators. Based on the encoding method, we develop the genetic operators in the GA: crossover and mutation.

- Crossover / recombination is a process of exchanging genetic information. We adopt one-point crossover operation; the crossover points are randomly set.
- Mutation operation. Any change in a gene is called a *mutation*; we use point mutation.

We propose the following GA-based shot placement algorithm:

1. Find a skeleton and all its endpoints. Take one of the endpoints as a starting point.
2. Randomly search all the points in the skeleton using the GA to find the location and size of the best shots as follows:
   (a) Generate randomly m (= 100) points from all the points (e.g., $M = 1000$) in the skeleton. The m points are the chromosomes. Set the crossover rate $p_c = .95$, the mutation rate $p_m = .05$, the desired Fit function, and the number n_g of generations.
   (b) Calculate the Fit of all the m points in the skeleton.
   (c) If the algorithm has run n_g steps, or if one of the m points satisfies the desired Fit, the GA stops (at this time, the best shot is chosen); else encode m points into the bit strings.
   (d) To do crossover and mutation operation to the m bitstrings, go to 2b.
3. Considering the rest of the target in whole as a new target, repeat Steps 1–2 until the rest of the target are fractions (at this time, all best shots are found).

Dose-Based Optimization

After we obtain the number, sizes and the locations of shots, we develop a dose-based optimization method.

We determine a functional form for the dose delivered at a point (i, j, k) from the shot centered at (x_s, y_s, z_s). The complete dose distribution can be calculated as a sum of contributions from all of the individual shots of radiation:

$$D(i,j,k) = \sum_{(s,w)\in S\times W} t_{s,w} D_{s,w}(x_s, y_s, z_s, i, j, k),$$

where $D_{s,w}(x_s, y_s, z_s, i, j, k)$ is the dose delivered to (i, j, k) by the shot of size w centered at (x_s, y_s, z_s) with a delivery duration of $t_{s,w}$. Since $D_{s,w}$ is a complicated (nonconvex) function, we approximate it by

$$D_{s,w}(x_s, y_s, z_s, i, j, k) = \sum_{i=1}^{2} \lambda_i \left(1 - \int_{-\infty}^{x} \frac{1}{\sqrt{\pi}} e^{-x^2}\, dx\right), \qquad \textbf{(4)}$$

where $x = (t - r_i)/\sigma_i$ and λ_i, γ_i, and σ_i are coefficients [Ferris and Shepard 2000].

To meet the requirement of matching specified isodose contours to target volume at the 50% isodose line, the optimization formulation should impose a constraint on the 50% isodose line that must surround the target. We impose strict lower and upper bounds on the dose allowed in the target, namely, for all $(i, j, k) \in T$, the dose $D(i, j, k)$ satisfies

$$0.5 \le D(i, j, k) \le 2. \qquad \textbf{(5)}$$

To meet the requirement (R2) to match specified dose-volume constraints of the target and critical organ, based on assumption (A3) (which sets out the size of the beam channel diameters), no more than n shots are to be used; so in each *card* (any section of the target) we have

$$\text{card}\big[\{(s, w) \in S \times W \mid t_{s,w} > 0\}\big] \le n. \qquad \textbf{(6)}$$

The value of tolerance doses of the normal tissue points is q, $D(i, j, k) < q$ for all (i, j, k). The number of shots n is no more than 15, so the tolerance doses of a specified normal tissue point should be $q = 15/201 = 7.46\%$, or

$$0 \le D(i, j, k) < q = 7.46\%, \qquad \textbf{(7)}$$

for all $(i, j, k) \in N$.

To meet requirement (R3) (keep the does at normal tissue below a certain level), based on assumption (A6) (which sets the does levels), the tolerance dose radio of the total dose deposited in the target and critical organ to the total dose delivered by a plan is

$$\frac{\sum_{(i,j,k)\in T} D(i, j, k)}{\sum_{(i,j,k)\in T\cup N} D(i, j, k)} \ge P, \qquad P \in [.25, .40]. \qquad \textbf{(8)}$$

We wish to satisfy constraint (C3) (at least 90% of the target volume must be covered). We set

V = the total volume of target;

V_s = the total effective dosage volume of the target whose dose value at the point is more than 0.5; and

f = the effective dosage rate, which satisfies the inequality

$$90\% \leq f = \frac{V_s}{V} \leq 100\%. \qquad \textbf{(9)}$$

The exposure time of each shot $t_{s,w}$ should be nonnegative:

$$t_{s,w} \geq 0. \qquad \textbf{(10)}$$

We introduce a binary variable $\delta_{s,w}$ that indicates whether shot s uses width w or not, i.e.,

$$\delta_{s,w} = \begin{cases} 1, & \text{if shot } s \text{ uses width } w; \\ 0, & \text{otherwise.} \end{cases}$$

Moreover, we have the constraints (C1) (no shots protrude outside the target), (C2) (shots do not overlap), and (C4) (as few shots as necessary).

Given all these constraints **(5)–(10)**, and based on the requirement (R4) (minimize dose to normal tissue), the goal is to minimize the dose outside of the target.

Also, to meet the requirement (R5) (minimize the dose gradient across the target volume), the treatment plan needs to be both conformal and homogeneous. It is easy to specify homogeneity in the models simply by imposing lower and upper bounds on the dose delivered to points in the target T. Typically, however, the imposition of rigid bounds leads to plans that are overly homogeneous and not conformal enough—that is, they provide too much dose outside the target. To overcome this problem, the notion of *underdose* (UD) is suggested in Ferris and Shepard [2000]. UD measures how much the delivered dose is below the prescribed dose on the target points. In our models, we either constrain UD to be less than a prespecified value or attempt to minimize the total UD.

In practical application, rather than calculating the dose at every point, it is easy to estimate accurately the total dose delivered by a plan based solely on the $t_{s,w}$ variables and other precalculated constants. An upper bound is also placed on the dose to the target. Given these constraints, the optimizer seeks to minimize the total underdosage in the target. A point is considered to be underdosed if it receives less than the prescribed isodose θ, which for the example formulation is assumed to be 1. We actually use the optimization process to model UD, which is constrained to be

$$\text{UD}(i,j,k)m = \max\big[0, 1 - D(i,j,k)\big]$$

at every point in the target. We can implement this construct using linear constraints

$$\theta \leq \text{UD}(i,j,k) + D(i,j,k), \qquad \textbf{(11)}$$

$$0 \leq \text{UD}(i,j,k) \qquad \textbf{(12)}$$

for all $(i, j, k) \in T$.

Our second minimization problem is

$$\text{Objective: } \min \sum_{(i,j,k) \in N} \text{UD}(i, j, k)$$

subject to the same constraints **(5)–(10)** as earlier plus **(11)–(12)**.

To meet the requirement (R6) (minimize maximum dose to critical volumes), we have the additional optimization problem

$$\text{Objective: } \min \sum_{(i,j,k) \in N} D(i, j, k) \text{ for all } (i, j, k) \in T \text{ for which } \delta_{s,w} = 0$$

subject to the same constraints **(5)–(10)** as earlier.

All of the formulations are based on the assumption that the neurosurgeon can determine a priori a realistic upper bound n on the number of shots needed. Several issues need to be resolved to create models that are practical, implementable, and solvable (in a reasonable time frame). Two main approaches are proposed in the literature [Ferris et al. 2002; 2003; Shepard et al. 2000; Ferris and Shepard 2000], namely mixed integer programming and nonlinear programming, to optimize simultaneously all of the variables.

Simulation Results and Model Testing

We developed an optimization package to implement the algorithms of our models in Matlab and perform numerous computer simulations using targets of different shapes and sizes.

To examine its correctness, we plot a dose-volume histogram for the four different helmets using **(4)**, as shown in **Figure 2**. The histogram depicts the fraction of the volume that receives a particular dose for the target volume. The fit is best for the small shots and decreases slightly in accuracy for the larger ones. The lines show the fraction of the target and critical organ that receives a particular dosage.

Generally speaking, the shape of the target is not too irregular, so we choose five typical shapes of the targets in different sizes. In **Figure 3a**, we illustrate the maximum section of a typical bean-shaped target, whose maximum dimension is 35 mm. Using the skeleton generation algorithm, we get the corresponding skeleton shown in **Figure 3b**. Then, we apply the GA-based shot placement algorithm, resulting in three shots for the target: one 14 mm helmet and two 8 mm helmets. The locations and sizes of the helmets in 2D are indicated in **Figure 3c**, while 3D shot placements are shown in **Figure 4**.

For this target, we also plot six different isodose lines: 30%, 40%, 50%, 60%, 70%, and 100% (**Figure 5**). The thick (red) line is the target outline, and the thin (black) line is the isodose line. In **Figure 5c**, the 50% isodose line covers all the points of the target, while in **Figure 5f** for the 100% isodose line, no point

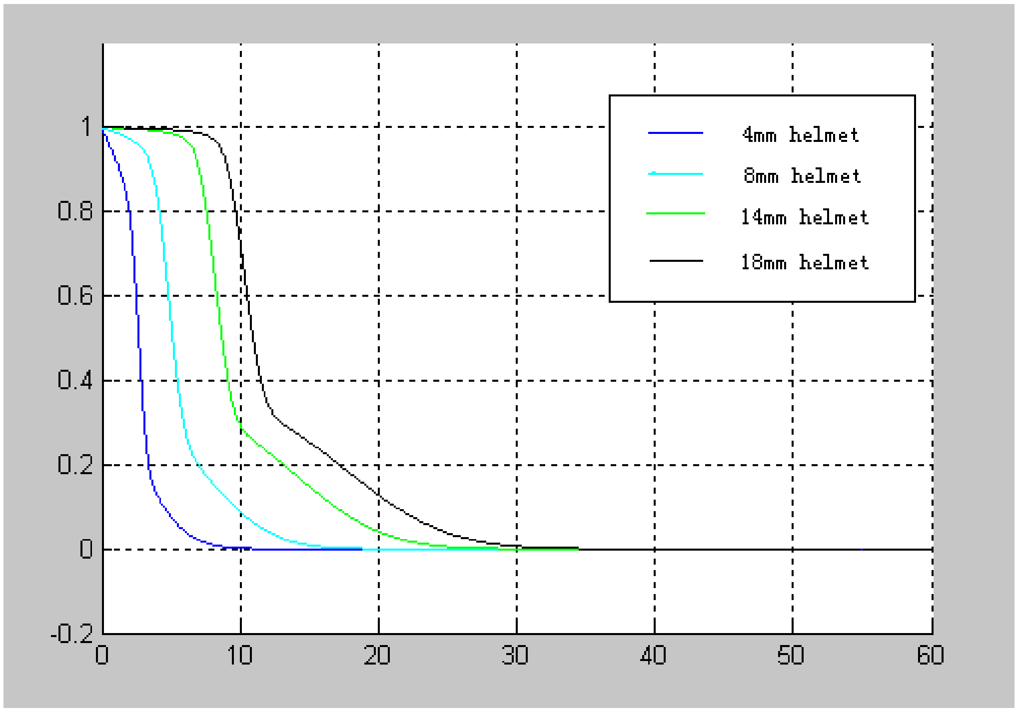


Figure 2. Dose–volume histograms for four different helmets.

Figure 3a. The maximum section of the target.

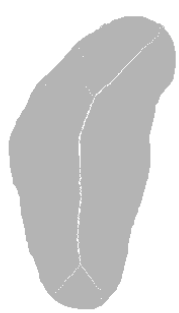


Figure 3b. The skeleton.

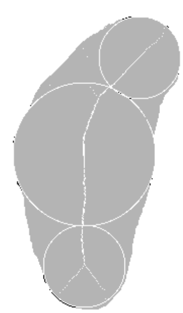


Figure 3c. The locations and sizes of the helmets, in 2D.

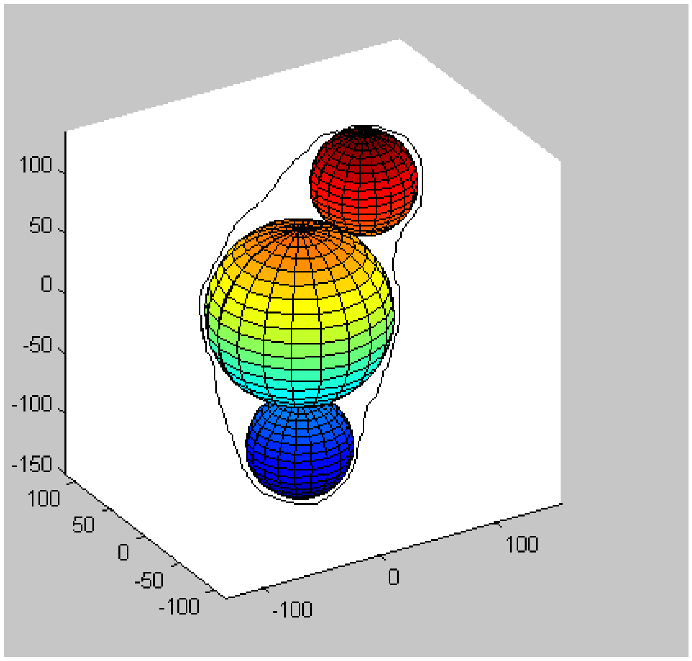


Figure 4. The 3D shot placements in the target.

of the shots exceeds the boundary of the target. We also present the 3D shot placements for four other target shapes in **Figure 6**.

The optimized plans for all of the five shapes of the targets are shown in **Table 1**, together with the minimum target doses and the percentage coverages.

Table 1.

Optimized plans for five targets.

Target (figure)	Maximum section width (mm)	Helmet sizes (mm)	Number of shots	Minimum target dose	Coverage (by isodose) 50%	80%	100%
6	35	18 4	1 5	0.51	100%	97%	90%
8a	26	8 4	4 3	0.52	100%	97%	88%
8b	20	14 8 4	1 1 1	0.52	100%	96%	92%
8c	10	4	6	0.45	99%	82%	69%
8d	8	4	2	0.67	100%	97%	57%

From all of the results, we know that the geometry-based heuristic with the GA optimization approach is a useful tool for assisting in the selection of the appropriate number of shots and helmet sizes. Also, they indicate that our model exceeds the predefined quality of the treatment planning.

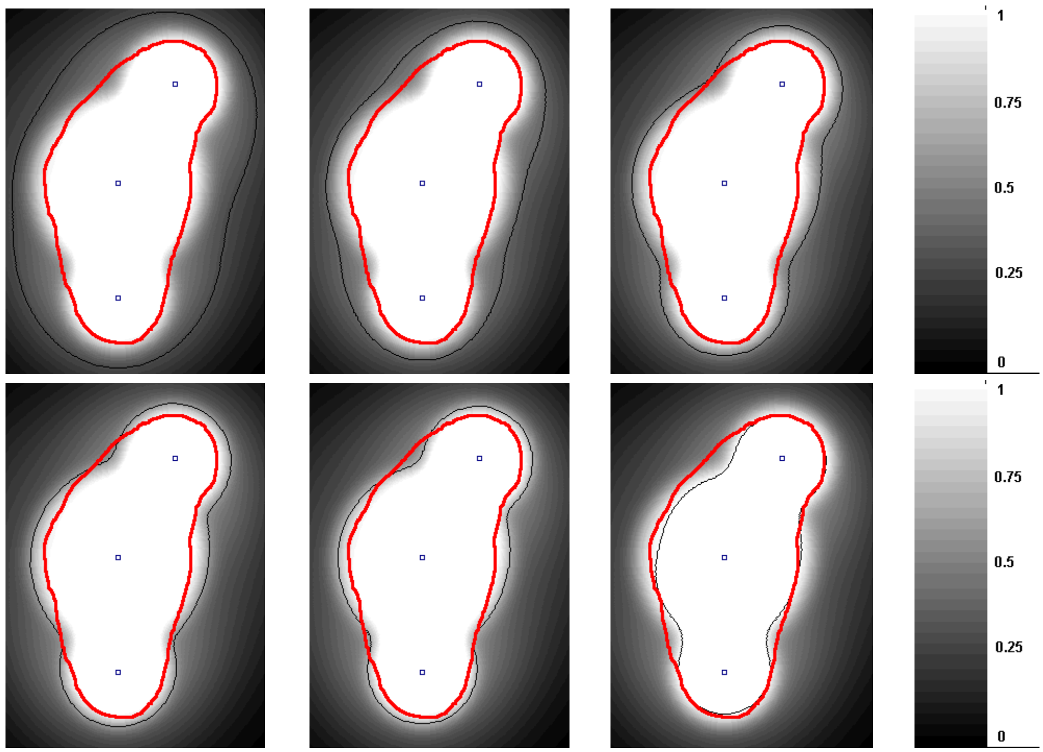


Figure 5. The specified isodose lines of different values: 30%, 40% 50%; 60%, 70%, 100%.

Sensitivity Analysis

- *Can the model be applied to sensitive structures?* Yes, by applying more dose constraints, such as an upper bound on either the mean dose or the maximum dose to the sensitive structures

- *Can we treat the tumor at an isodose level other than 50%?* In **Figure 5**, with a lower isodose line, the dose outside of the target volume decreases rapidly, resulting in a reduction in the integral dose to normal tissue. With a higher isodose line, the isodose line cannot cover all the points of the target. For the nodular regions or sensitive organs, the higher isodose coverage level should be specified.

- *The outstanding question from an optimization viewpoint is global vs. local optimality. How about our model?* First, the GA-based shot placement algorithm can ensure that the generating fraction is the smallest one after every shot is placed on the target. For the whole target, it also minimizes the sum of all fractions.

- *When we have several comparably optimization schemes for shot placement, how do we choose the best one?* For example, for a 10 mm-diameter sphere target, we have two comparably optimization schemes, as shown in **Table 1**. The first places one 8-mm shot and the second places six 4-mm shots. If we consider the treatment merely as a sphere-packing problem, the better choice is the

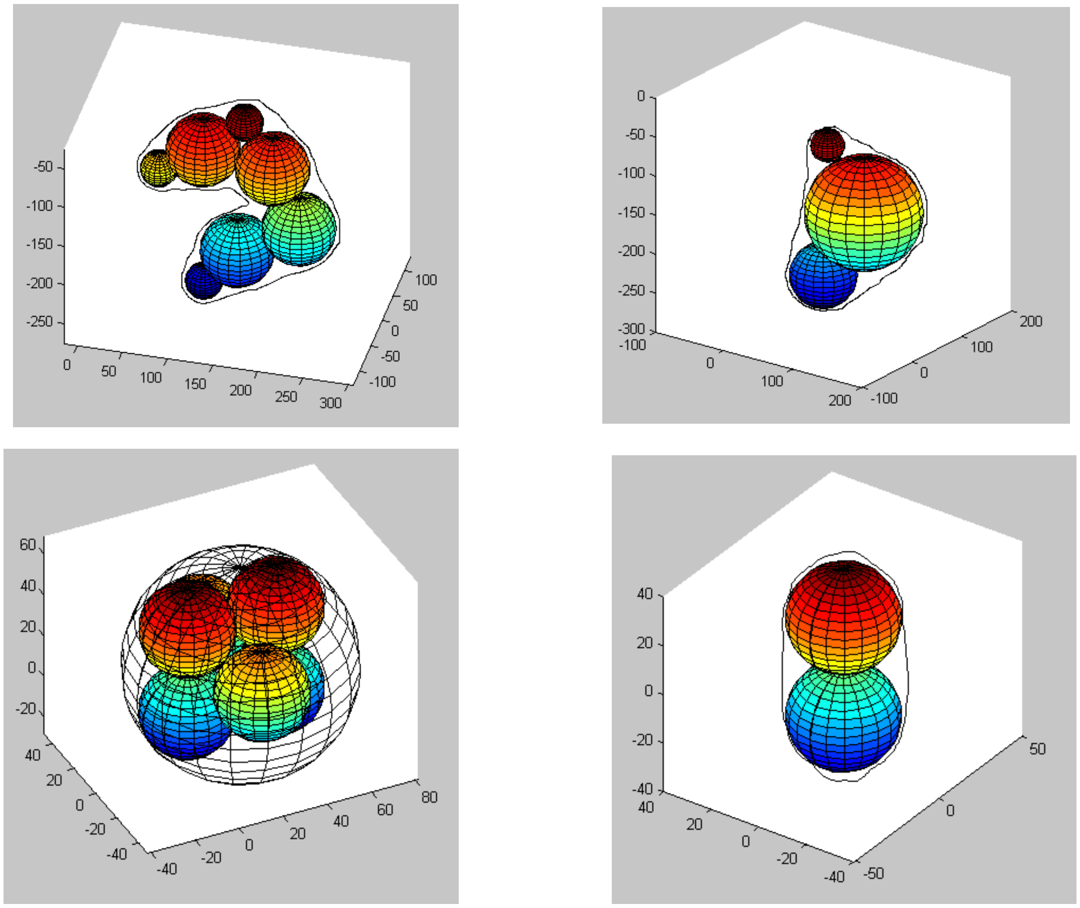


Figure 6. Shot placement in four targets.

first one. However, in practical treatment, we should consider the diffuse regions where no shot is irradiated. Under the first plan, inn these regions the sum of the dose value at a point is more than 1, resulting in increasing the total effective dosage; so we adopt the second plan.

- *Will there be any points outside the tumor whose dose value is greater than 1?* In **Figure 7**, though the shots have not protruded outside the target (constraint (C1)), some points outside the tumor overdose. This occurs due to the very irregular shape of the target, which is not avoidable. In this case, we should choose how to choose an optimization planning under some constraints.

- *How about the robustness of our model?* For the many cases optimized thus far, high-quality dose distributions have been obtained in all cases.

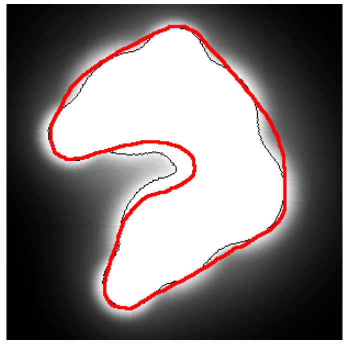


Figure 7. The 100% isodose of the target in **Figure 6a**.

Strengths and Limitations

Strengths

- Our optimization-based automated approach generates more-uniform and better treatment plans in less time than is currently used.
- The geometry-based approach is based on skeletonization ideas from computational graphics, which can speed the process of shot placement.
- The GA-based shot placement algorithm can guide the planner in selecting the number of shots of radiation and the appropriate collimator helmet sizes, it can quickly place a shot, and it can ensure global and local optimality simultaneously.
- The model parameters can be tuned to improve solution speed and robustness.
- The graphical interface is an intuitive way to demonstrate the isodose curve and the treatment effects of the planning.

Limitations

- The skeleton is a key factor for the effectiveness of the algorithm; we should seek better methods to determine it.
- Whether our model can handle very irregular targets needs to be examined.
- We use the function in Ferris and Shepard [2000] to approximate the dose calculation. Other methods of dose calculation should be examined.
- There is no guarantee that there is not a better treatment plan. Some more-intelligent algorithms, such as a neural network-based dynamic programming algorithm, could be considered.
- We have not tested our model on actual patient data.

Conclusions

From the simulation results, we know that the geometry-based heuristic with the GA optimization approach is a useful tool in the selection of the appropriate number of shots and helmet sizes. Our approach is sufficiently robust and effective to be used in practice.

In future work, we may fit ellipsoids instead of spheres, since some researchers have commented that the dose is skewed in certain directions.

References

Cho, P.S., H.G. Kuterdem, and R.J. Marks. 1998. A spherical dose model for radiosurgery plan optimization. *Physics in Medicine and Biology* 43 (10): 3145–3148.

Ferris, M.C., J.-H. Lim and D.M. Shepard. 2002. An optimization approach for radiosurgery treatment planning. *SIAM Journal on Optimization* 13 (3): 921–937. `ftp://ftp.cs.wisc.edu/pub/dmi/tech-reports/01-12.pdf` .

________ . 2003. Radiosurgery treatment planning via nonlinear programming. *Annals of Operations Research* 119 (1): 247–260. `ftp://ftp.cs.wisc.edu/pub/dmi/tech-reports/01-01.pdf` .

Ferris, M.C., and D.M. Shepard. 2000. Optimization of gamma knife radiosurgery. In *Discrete Mathematical Problems with Medical Applications*, vol. 55 of DIMACS Series in Discrete Mathematics and Theoretical Computer Science, edited by D.-Z. Du, P. Pardolas, and J. Wang, 27–44. Providence, RI: American Mathematical Society. `ftp://ftp.cs.wisc.edu/pub/dmi/tech-reports/00-01.pdf`

Goldberg, D.E. 1989. *Genetic Algorithms: In Search, Optimization and Machine Learning*. Reading, MA: Addison-Wesley.

Leichtman, Gregg S., Anthony L. Aita, and H. Warren Goldman. 2000. Automated gamma knife dose planning using polygon clipping and adaptive simulated annealing. *Medical Physics* 27 (1): 154–162.

Liu, J.-F., and R.-X. Tang. 1997. Ball-packing method: A new approach for quality automatic triangulation of arbitrary domains. In *Proceedings of the 6th International Meshing Roundtable*, 85–96. Sandia, NM: Sandia National Laboratories.

Man, K.F., K.S. Tang, S. Kwong, and W.A. Halang. 1997. *Genetic Algorithms for Control and Signal Processing*. New York: Springer-Verlag.

Shepard, D.M., M.C. Ferris, R. Ove, and L. Ma. 2000. Inverse treatment planning for gamma knife radiosurgery. *Medical Physics* 27 (12): 2748–2756.

Shu, Huazhong, Yulong Yan, Limin Luo, and Xudong Bao. 1998. Three-dimensional optimization of treatment planning for gamma unit treatment system. *Medical Physics* 25 (12): 2352–2357.

Wu, J.Q. 2000. Sphere packing using morphological analysis. In *Discrete Mathematical Problems with Medical Applications*, vol. 55 of DIMACS Series in Discrete Mathematics and Theoretical Computer Science, edited by D.-Z. Du, P. Pardolas, and J. Wang, 45–54. Providence, RI: American Mathematical Society.

________, and J.D. Bourland. 2000a. A study and automatic solution for multi-shot treatment planning for the gamma knife. *Journal of Radiosurgery* 3 (2): 77–84.

________. 2000b. 3D skeletonization and its application in shape based optimization. *Computers in Medical Imaging* 24 (4): 243–251.

________, and R.A. Robb. 1996. Fast 3D medial axis transformation to reduce computation and complexity in radiosurgery treatment planning. In *Medical Imaging 1996: Image Processing, Proceedings of the SPIE*, vol. 2710, edited by Murray H. Loew and Kenneth M. Hanson, 562–571.

Zhang, Pengpeng, David Dean, Andrew Metzger, and Claudio Sibata. 2001. Optimization of gamma knife treatment planning via guided evolutionary simulated annealing. *Medical Physics* 28 (8): 1746–1752.

Zhou, Y., A. Kaufman, and A.W. Toga. 1998. Three dimensional skeleton and centerline generation based on an approximate minimum distance field. *Visual Computers* 14: 303–314.

2004: The Fingerprints Problem

It is a commonplace belief that the thumbprint of every human who has ever lived is different.

Develop and analyze a model that will allow you to assess the probability that this is true.

Compare the odds (that you found in this problem) of misidentification by fingerprint evidence against the odds of misidentification by DNA evidence.

Comments

The Outstanding papers were by teams from Harvey Mudd College, University College Cork (Ireland), and University of Colorado. Their papers and accompanying commentaries were published in *The UMAP Journal* 25 (3) (2004): 215–280. The Ben Fusaro Award went to a team from Central Washington University.

Problem Origin

The problem was contributed by Michael Tortorella (Rutgers University).

Practitioner's Commentary

Mary Beeton, Automated Fingerprint Identification System Technician for the Durham Regional Police Service in Oshawa, Ontario, Canada, commented from the perspective of someone who examines more than 125,000 fingerprints each year.

Despite the oversimplification in each model from unrealistic assumptions such as

- minutiae occur at uniform rates along a particular ridge,
- only two types of minutiae exist,
- all fingerprints are perfect rolled prints, and
- ridge widths are consistent, so that pores and edge shapes are not significant,

each of the teams' conclusions support fingerprint uniqueness.

In 1993, a U.S. Supreme Court, in a ruling for the civil case *Daubert v. Merrell Dow Pharmaceuticals, Inc.*, 509 U.S. 579 (1993), outlined specific "Daubert" criteria to be used by trial judges in assessing the admissibility of scientific evidence. In 1999, the Assistant Federal Defender for the State of Pennsylvania challenged the admissibility of fingerprint evidence, based on the "Daubert" criteria.

How can Friction Ridge Identification Specialists claim certainty? Champod et al. [2004] argue that positive identification "is based on inductive reasoning" and, therefore, "must be probabilistic." However, others suggest that the probability is so small that it can be disregarded and hence the latent print examiner's conclusion of 100% certainty is acceptable.

There are inherent risks in bringing a probability model into a courtroom. Regarding one DNA "match" case, "The odds of the arrestee's DNA being wrongly matched against that of the crime scene were said to be one in 37 million" [Moenssens 2000]. Moenssens believes that it is a common misunderstanding among lawyers, judges, lay persons, and police that when a DNA "match" is reported with odds of one in 37 million that a like match in the DNA pattern exists once in 37 million people. Moenssens adds, "the calculated frequency is an estimate, and can be off by an order of magnitude in either direction."

Whether or not fingerprint features are recognized, ignored, or given any significance can be seriously affected by any distortion present in the fingerprint. Another factor is that the majority of crime scene fingerprints contain only 20% of the information found in a rolled inked fingerprint.

Author-Judge's Commentary

A reasonable interpretation of the relevant question could be:

> *Determine the probability that there have never been two identical fingerprints, given the capability of the technology used to determine "different."*

You get very quickly into deep—even philosophical—questions.

The first step in developing a model based on this question is to *define*:

- "fingerprint,"
- the probability space in which this experiment is conducted, and
- "distinct."

"Distinct" depends on resolution.

The definition of "fingerprint" is wrapped up in the definition of the probability space, because most teams made assumptions about the minimum spacing between ridges that could possibly occur. This assumption is based on empirical evidence and is the first step down a road leading to consideration of only a finite number of potential fingerprints.

As always, interpretation is the key to success in modeling problems. The first key was to understand that the word "fingerprint," in addition to its usual semantic or prose usage, must be given a *mathematical meaning* in the context of a model.

With a few assumptions, a model is possible. Assume that fingerprints (the actual skin patterns) are assigned at birth, at random from a pool of potential fingerprints. If we assume that the pool contains $N \gg n$ elements and selection is made on an equally-likely basis, then the probability that there are no two fingerprints alike is the solution to a birthday problem with n people and N birthdays; namely, the probability of no match (denoted $Q_1 = 1 - P_2$ in Weisstein [2004]) is given by

$$P(\text{no match}) = \frac{N}{N}\frac{N-1}{N}\cdots\frac{N-n+1}{N} = \frac{N!}{(N-n)!N^n} \approx e^{-n(n-1)/2N},$$

which for fixed n is asymptotic to 1 as $N \to \infty$.

Further definitions for "fingerprint" shrink the pool of possible "fingerprints," that is, restrict N so that there is a chance that the probability of no match will be less than 1.

The total number of possible distinct fingerprints implied is $2^{600} \approx 10^{181}$. The number of people who have ever lived is about 1.06×10^{11}; so, assuming that all 2^{600} patterns are equally likely, the probability that no two persons who have ever lived have the same fingerprint is approximately $1 - 10^{-159}/2$ (this latter computed from the "birthday problem" with $1.06 \times 10^{1}1$ people and 2^{600} possible "birthdays," a point that many teams missed).

It is easy to poke holes in this model. Empirically, it is clear that

- not every grid square has the same probability of containing a minutia,
- stochastic independence of the presence or absence of minutiae from grid square to grid square is not reasonable, and
- there are several different types of minutiae.

When such assumptions based on empirical observation are introduced, the modeler should attempt to bound the answers using a range of possible reasonable values for the inputs, because sampling error could affect the assumptions. Once could argue that sampling error should be negligible in drawing inferences from a database containing millions of records, like most fingerprint databases, but most teams did not address this issue in any way.

Finally, the problem asks for comparison of the computed probability with the probability of misidentification by DNA evidence, a topic much in the public eye in the last decade. Some teams ignored this requirement. Others quoted popular anecdotes concerning the DNA misidentification probability. In the latter case, teams would be advised to bolster their contentions with at least one legitimate citation.

Further Commentary

Paul Campbell (Beloit College) offered his perspective as long-time editor of the Outstanding papers for *The UMAP Journal*.

Some problems in the MCM and the ICM arise from special situations, and it is not clear that specific ideas from the solution papers could have any application beyond the original setting:

- Emergency Facilities Location Problem (1986),
- Parking Lot Problem (1987),
- Midge Classification Problem (1989),
- Helix Intersection Problem (1995),
- Velociraptor Problem (1997),
- Lawful Capacity Problem (1999),
- Bicycle Wheel Problem (2001),
- Wind and Waterspray Problem (2002),
- Gamma Knife Treatment Problem (2003), and
- Stunt Person Problem (2003).

Other problems arise from situations that society faces chronically but have no urgency, yet the solution papers provide valuable ideas that could be put into practice:

- Salt Storage Problem (1987),
- College Salaries Problem (1995),
- Contest Judging Problem (1996),
- Discussion Groups Problem (1997),
- Grade Inflation Problem (1998), and
- Quick Pass Problem (2004).

Finally, some problems touch on issues of immediate concern and the solution papers offer important insights:

- Emergency Power-Restoration Problem (1992),
- Asteroid Impact Problem (1999),
- Hurricane Evacuation Problem (2001),
- Airline Overbooking Problem (2002),
- IT Security Problem (ICM 2003), and
- Airport Security Problem (2004).

Perhaps no problem has been as aptly timed, however, as the Fingerprints Problem. The Outstanding papers and the commentaries refer to very recent questioning in U.S. courts of the reliability of fingerprint evidence.

Fingerprint identification is not solely a scientific enterprise but takes place in an environment where various sources of human error can prevail: fraud, incompetence, and errors of various kinds. Prof. Campbell cited instances of each in recent events. He also emphasized two philosophical questions with important ramifications:

- the assumption (by all teams) that all the many fingerprints are equally (un)likely. "We do not know how often different people's prints may significantly resemble one another, or how good examiners are at distinguishing between such prints" [Mnookin 2004].
- the status of scientific truth and of evidence obtained by technical means. Can scientific evidence yield certainty? Should scientific evidence be regarded as more reliable than other evidence? Is it more reliable?

In the final analysis, declaring a match of two fingerprints is a subjective decision, made by a human being based on training, experience, and all of the circumstances involved in the comparison.

References

Begley, Sharon. 2004. Despite its reputation, fingerprint evidence isn't really infallible. *Wall Street Journal* (4 June 2004) B1.

Champod, Christophe, Chris Lennard, Pierre Margot, and Milutin Stoilovic. 2004. *Fingerprints and Other Ridge Skin Impressions*. Boca Raton, FL: CRC Press.

Mnookin, Jennifer L. 2004. The Achilles' heel of fingerprints. *The Washington Post* (29 May 2004): A27. `http://www.washingtonpost.com/wp-dyn/articles/A64711-2004May28.html` .

Moenssens, Andre A. 2000. Mistaken DNA identification? What does it mean? Updated 20 October 2000. `http://www.forensic-evidence.com/site/EVID/EL_DNAerror.html` .

Weisstein, Eric. 2004. Birthday problem. `http://mathworld.wolfram.com/BirthdayProblem.html` .

The Myth of "The Myth of Fingerprints"

Steven G. Amery
Eric Thomas Harley
Eric J. Malm
Harvey Mudd College
Claremont, CA

Advisor: Jon Jacobsen

Summary

For over a century, fingerprints have been an undisputed personal identifier. Recent court rulings have sparked interest in verifying uniqueness of fingerprints.

We seek to determine precisely the probability of duplicate fingerprints. Our model of fingerprint structure must achieve the following objectives:

- **Topological structure** in the print, determined by the overall flow of ridges and valleys, should be described accurately.
- **Fine detail**, in the form of ridge bifurcations and terminations, must also be characterized accurately.
- **Intrinsic uncertainties**, in our ability to reproduce and measure fingerprint data, must be considered.
- **Definite probabilities** for specified fingerprint configurations must be calculated.

We place special emphasis on meeting the modeling criteria established by Stoney and Thornton [1986] in their assessment of prior fingerprint models.

We apply our model to the conditions encountered in forensic science, to determine the legitimacy of current methodology. We also compare the accuracies of DNA and fingerprint evidence.

Our model predicts uniqueness of prints throughout human history. Furthermore, fingerprint evidence can be as valid as DNA evidence, if not more so, although both depend on the quality of the forensic material recovered.

Introduction

What is a Fingerprint?

A fingerprint is a two-dimensional pattern created by the friction ridges on a human finger [Beeton 2002]. Such ridges are believed to form in the embryo and to persist unchanged through life. The physical ridge structure appears to depend chaotically on factors such as genetic makeup and embryonic fluid flow [Prabhakar 2001]. When a finger is pressed onto a surface, the friction ridges transfer to it (via skin oil, ink, or blood) a representation of their structure.

Fingerprints have three levels of detail [Beeton 2002]:

1. Overall ridge flow and scarring patterns, insufficient for discrimination.
2. Bifurcations, terminations, and other discontinuities of ridges. The pairwise locations and orientations of the up to 60 such features in a full print, called *minutiae,* provide for detailed comparison [Pankanti et al. 2002].
3. The width of the ridges, the placement of pores, and other intraridge features. Such detail is frequently missing from all but the best of fingerprints.

Fingerprints as Evidence

The first two levels have been used to match suspects with crime scenes, and fingerprint evidence was long used without major challenge in U.S. courts [OnIn.com 2003]. In 1993, however, in *Daubert v. Merrill Dow Pharmaceutical*, the U.S. Supreme Court set standards for “scientific” evidence [Wayman 2000]:

1. The theory or technique has been or can be tested.
2. The theory or technique has been subjected to peer review or publication.
3. The existence and maintenance of standards controlling use of the technique.
4. General acceptance of the technique in the scientific community.
5. A known potential rate of error.

Since then, there have been challenges to fingerprint evidence.

Individuality of Fingerprints

Francis Galton [1892] divides a fingerprint into squares with a side length of six ridge periods and estimates that he can recreate the ridge structure of a missing square with probability $\frac{1}{2}$. Assuming independence of squares and introducing additional factors, he concludes that the probability of a given fingerprint occurring is 1.45×10^{-11}. Pearson refines Galton's model and finds a probability of 1.09×10^{-41} [Stoney and Thornton 1986].

Osterburg [1977] extends Galton's approach by dividing a fingerprint into cells that can each contain one of 12 minutia types. Based on independence among cells and observed frequencies of minutiae, he finds the probability of a configuration to be 1.33×10^{-27}. Sclove [1979] extends Osterburg's model to dependencies among cells and multiple minutiae in a single cell.

Stoney and Thornton [1986] charge that these models fail to consider key issues completely: the topological information in level-one detail; minutiae location, orientation, and type; normal variations in fingerprints; and number of positions considered. We try to correct some of these omissions.

Our Model: Assumptions and Constraints

Assumptions

- **Fingerprints are persistent:** they remain the same throughout a person's lifetime. Galton [1892] established this fact, in recent times verified from the processes of development of dermal tissues [Beeton 2002].

- **Fingerprints are of the highest possible quality,** without damage from abrasion and injury.

- **The pattern of ridges has some degree of continuity and flow.**

- **The ridge structure of a fingerprint is in one of five categories:** Arch, Left Loop, Right Loop, Tented Arch, or Whorl, employed in the automatic classification system of Cappelli et al. [1999] (derived from those of the FBI and Watson and Wilson [1992]). Each category has a characteristic ridge flow topology, which we break into homogeneous domains of approximately unidirectional flow. While Cappelli et al. [1999] raise the issue of "unclassifiable" prints, and they and Marcialis et al. [2001] confuse classes of ridge structures, we assume that such ambiguities stem from poor print quality.

- **Fingerprints may further be distinguished by the location and orientation of minutiae relative to local ridge flow.** Stoney and Thornton [1986] argue that the ridges define a natural coordinate system, so the location of a minutia can be specified with a ridge number and a linear measure along that ridge. Finally, minutiae have one of two equally likely orientations along a ridge.

- **Each minutia can be classified as a bifurcation, a termination, or a dot** (**Figure 1**) [Pankanti et al. 2002; Stoney and Thornton 1986]. Though Galton [1892] identifies 10 minutia structures and others find 13 [Osterburg et al. 1977], we can ignore these further structures (which are compositions of the basic three) because of their low frequency [Osterburg et al. 1977].

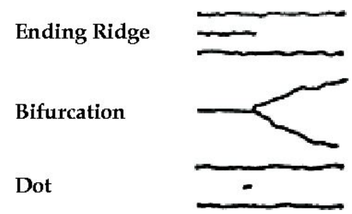


Figure 1. The three basic minutiae types (from Galton [1892]). We refer to ending ridges as *terminations.*

- **A ridge structure produces an unambiguous fingerprint, up to some level of resolution.** A ridge structure can vary in print representations primarily in geometric data, such as ridge spacing, curvature, and location of minutiae [Stoney and Thornton 1986]. Topological data—ridge counts, minutiae orientation, and ordering—are robust to such variation and are replicated consistently.

  A more serious consideration is connective ambiguities, such as when a given physical minutia is represented sometimes as a bifurcation and sometimes as a termination. But our highest-quality assumption dictates that such ambiguity arise only where the physical structure itself is ambiguous.

- **Location and orientation of minutiae relative to each other are independent,** though Stoney and Thornton [1986] find some dependency and Sclove [1979] model such dependency in a Markov process.

- **Ridge widths are uniform throughout the print and among different prints, and ridge detail such as pores and edge shapes is not significant.** While ridge detail is potentially useful, we have little data about types and frequencies.

- **Frequencies of ridge structure classes and configurations and minutiae types do not change appreciably with time.** We need this invariance for our model's probabilities to apply throughout human history.

Constraints Implied by Assumptions

- Our model must consider ridge structure, relative position, orientation, and type of minutiae.

- Locations of minutiae must be specified only to within some uncertainty dependent on the inherent uncertainty in feature representation.

Model Formulation

We examine a hierarchy of probabilities:

- that the given class of ridge structure occurs,

- that the ridge structure occurs in the specified configuration of ridge flow regions, and
- that minutiae are distributed as specified throughout the regions.

We further break this last probability down into a composition of the following region-specific probabilities:

- that a region contains the specified number of minutiae,
- that the minutiae in this region follow the specified configuration, and
- that the minutiae occur with the specified types and orientations.

Probability of Ridge Structure Class

To each of the five classes of ridge structures (Arches, Left and Right Loops, Tented Arches, and Whorls), we associate a probability of occurrence (ν_A, ν_L, ν_R, ν_T, ν_W), which we estimate from observed frequency in the population.

Probability of Ridge Structure Configuration

Each print is partitioned into regions in which the overall flow is relatively unidirectional, and the class of the print is determined from five prototypical masks characteristic of ridge-structure classes (**Figure 2**) [Cappelli et al. 1999]. The variations of flow region structure within each class then depend on parameters for the class. For example, the ridge structure of a Loop print can be determined from the locations of the triangular singularity and the core of the loop (**Figure 3**). To determine the probability of a particular region configuration, we determine the probability that the associated parameters occur.

Because of uncertainty in the parameters, we discretize the parameter space at the fundamental resolution limit δ_1 (subscript indicates feature level). We use independent Gaussian distributions about the mean values of the parameters.

We now detail the parameter spaces for each ridge-structure class. The use of the prototypes requires an $X \times Y$ region within the print.

Arch

The parameters for the Arch consist of the Cartesian coordinates (x, y) of the lower corner of the left region, the height h of the central corridor, and the four angles $\theta_1, \theta_2, \theta_3, \theta_4$ at the inner corners of the left and right regions. We consider as fixed the width b of the central corridor. The ratio of the resolution limit δ_1 to the mean length of a typical segment determines the uncertainty in the angular measurement of that segment.

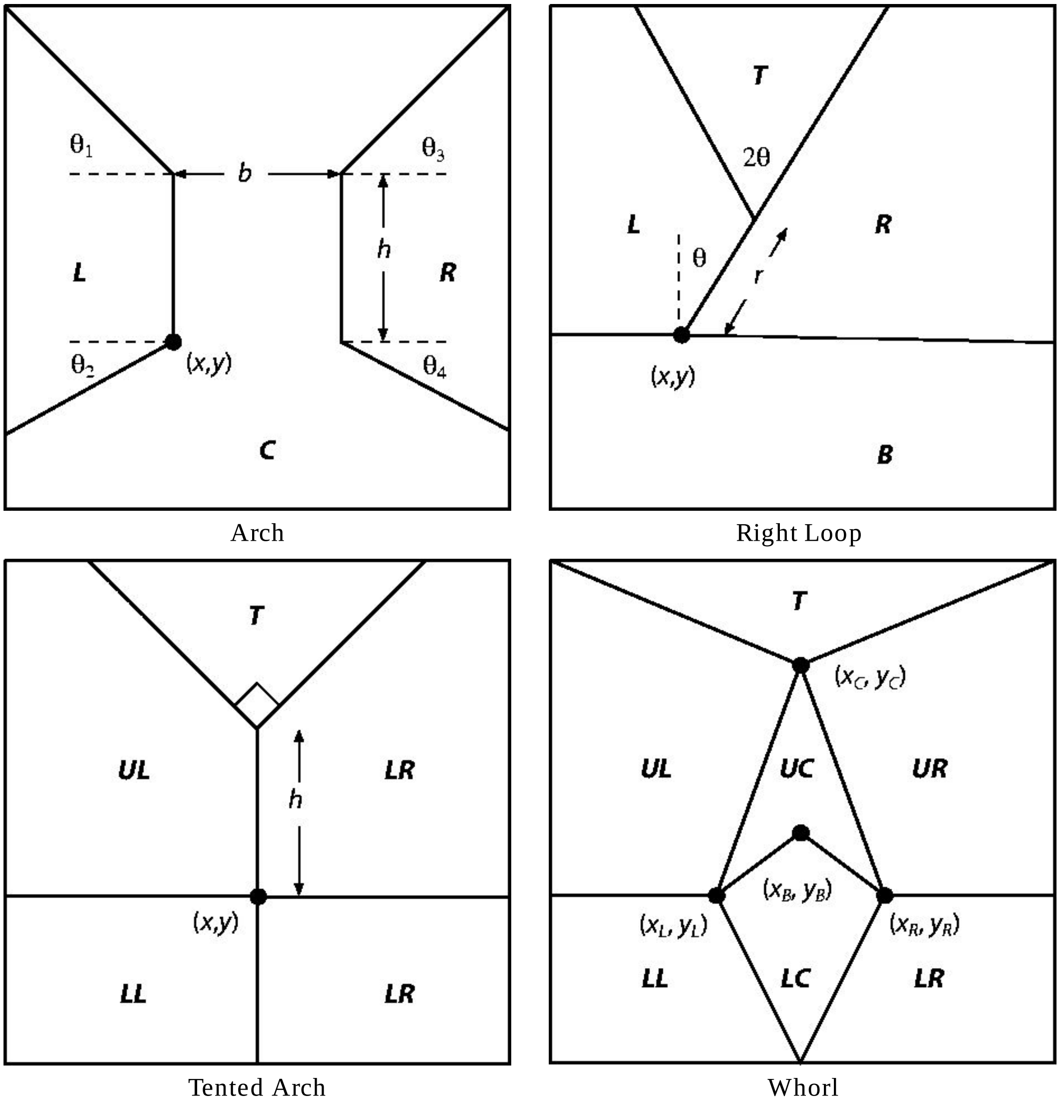


Figure 2. The prototypical region structures and parameters for each ridge structure class, derived from the masks in Cappelli et al. [1999].

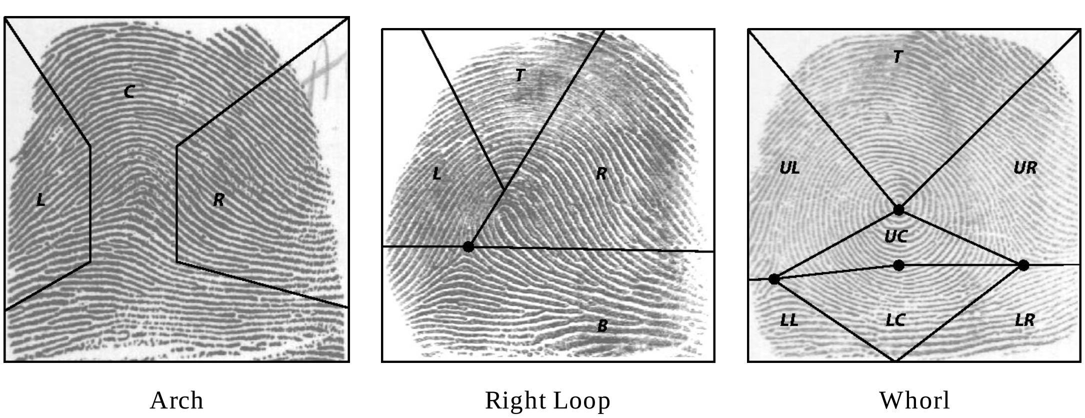


Figure 3. The prototypical region structures applied to an Arch, a Right Loop, and a Whorl.

Loops, Left and Right

Since Left and Right Loops are identical except for a horizontal reflection, we use the same parameter space for both classes. The two principal features are the position (x, y) of the triangular singularity outside the loop and the distance r and angle θ of the core of the loop relative to this singularity.

Tented Arch

The major structure is the arch itself; the parameters are the position (x, y) of the base of the arch and the height h of the arch.

Whorl

The Whorl structure has four major features: the center of the whorl, (x_C, y_C); the base of the whorl, (x_B, y_B); and the triangular singularities to the left and right of the base of the whorl, at $(x_L, y_{:})$ and (x_R, y_R). We assume that the center and the base lie between the two singularities, so that $x_L \leq x_C$ and $x_B \leq x_R$, and that the base lies above the singularities, so that $y_B \geq y_L$ and $y_B \geq y_R$.

Probabilities of Intraregion Minutiae Distribution

Since the geometry of a region is uniquely determined by the configuration parameters, we can divide each unidirectional flow region into parallel ridges. We can represent the ridge structure of the region as a list of ridge lengths.

We assume a fundamental limit δ_2 to resolution of the position of minutiae along a ridge and divide a ridge into cells of length δ_2, in each of which we find at most one minutia. The probability $P_{TC}(n, l, k)$ that the nth ridge in the partition, with length l, has a particular configuration of k minutiae is

$$P_{TC}(n, l, k, \dots) = P_p(n, k, l) P_c(n, k, l) P_{to}(\{k_i, p_i, o_i\}),$$

where P_p is the probability that k minutiae occur on this ridge, P_c the probability that these k minutiae are configured in the specified pattern on the ridge, and P_{to} the probability that these minutiae are of the specified types and orientations, indexed by i and occurring with type probability p_i and orientation probability o_i.

Probability of Minutiae Number

Under the assumption that minutiae occur at uniform rates along a ridge, we expect a binomial distribution for the number of minutiae on the ridge. Denote the linear minutiae density on ridge n by $\lambda(n)$. The probability that a minutia occurs in a given cell of length δ_2 is $\delta_2 \lambda(n)$. Thus, the probability that k minutiae occur is

$$P_p(n, k, l, \lambda) = \binom{l/\delta_2}{k} (\delta_2 \lambda)^k (1 - \delta_2 \lambda)^{l/\delta_2 - k}.$$

Probability of Minutiae Configuration

Assuming that all configurations of k minutiae are equally likely along the ridge, the probability of the specified configuration is

$$P_c(n,k,l) = \frac{1}{\binom{l/\delta_2}{k}}.$$

Probability of Minutiae Type and Orientation

The probability that minutiae occur with specified types and orientations is

$$P_{to}(\{k_i, p_i, o_i\}) = \prod_i p_i^{k_i} o_i^{k_i}.$$

Applying our assumption that the only level-two features are bifurcations, terminations, and dots, and that orientations are equally likely and independent along the ridge, this expression reduces to

$$P_{to} = p_b^{k_b} p_t^{k_t} p_d^{k_d} \, \frac{1}{2^{k_b+k_t}},$$

with $k_b + k_t + k_d = k$. Then the total probability for the ridge configuration is

$$P_{TC}(n,l,k,\lambda,\{k_i,p_i,o_i\}) = (\delta_2\lambda)^k (1-\delta_2\lambda)^{l/\delta_2 - k} p_b^{k_b} p_t^{k_t} p_d^{k_d} \, \frac{1}{2^{k_b+k_t}}.$$

The total probability that minutiae are configured as specified through the entire print is then product of the P_{TC}s for all ridges in all domains, since we assume that ridges develop minutiae independently.

Applying the assumption that λ and other factors do not depend on n and are hence uniform throughout the print, we can collapse these multiplicative factors to an expression for the configuration probability of the entire print:

$$P_{TC}^{\text{global}} = (\delta_2\lambda)^K (1-\delta_2\lambda)^{L/\delta_2 - K} p_b^{K_b} p_t^{K_t} p_d^{K_d} \, \frac{1}{2^{K_b+K_t}}.$$

Here K is the total number of minutiae in the print, K_i the number of type i, and L is the total linear length of the ridge structure in the print. The length L is determined only by the total area XY of the print and the average ridge width w and is therefore independent of the topological structure of the print.

Parameter Estimation

For parameters in our model, we use published values and estimates based on the NIST-4 database [Watson and Wilson 1992].

Level-One Parameters

All lengths are in millimeters (mm); angles are in radians or in degrees.

- **Level-one spatial resolution limit** δ_1: Cappelli et al. [1999] discretize images into a 28×30 grid to determine level-one detail. From these grid dimensions, the physical dimensions of fingerprints, and the assumption of an uncertainty of three blocks for any level-one feature, we estimate $\delta_1 = 1.5$.
- **Level-one angular resolution limit** δ_θ: Taking $X/2 = 5.4$ (determined below) as a typical length scale, we have $\delta_\theta = \delta_1/5.4 = 0.279$ radians.
- **Ridge structure class frequencies** ν_A**,** ν_L**,** ν_R**,** ν_T**, and** ν_W: We use the estimates in Prabhakar [2001] (**Table 1**).

Table 1. Relative frequencies of ridge structure classes (from Prabhakar [2001]).

ν_A	ν_L	ν_R	ν_T	ν_W
0.0616	0.1703	0.3648	0.0779	0.3252

- **Thumbprint width** X **and height** Y: Examining thumbprints from the NIST-4 database and comparing them with the area given by Pankanti et al. [2002], we conclude that a width that covers the majority of thumbprints is 212 pixels in the 500 dpi images, a physical length of 10.8 mm. Similarly, $Y = 16.2$ mm.
- **Arch parameters** (x, y)**,** h**,** b**,** θ_1**,** θ_2**,** θ_3**, and** θ_4: We restrict the parameter space for (x, y) to the lower half of the thumbprint with horizontal margins of length b. We estimate $b = 2.5$ from examination of Arch fingerprints in the NIST database and from Cappelli et al. [1999]. This estimate places $x \in (0, 8.3)$ and $y \in (0, 5.6)$. The mean for (x, y), which we need to describe the distribution of (x, y), is then $(4.2, 2.8)$. We estimate that x and y both have a standard deviation of 0.7. We assume that θ_1, ... ,θ_4 are all between $0°$ and $45°$ with mean $22.5°$ and standard deviation $5.13°$.
- **Loop parameters** (x, y)**,** θ**, and** r: For a left loop (a right loop is reflection of this), (x, y) must lie in the bottom left quadrant and the mean coordinate pair is $(2.7, 2.8)$. Additionally, we restrict θ to lie between $15°$ and $75°$, which allows us to estimate the mean θ as $45°$ with a standard deviation of $15°$. We estimate that r must be greater than 0 and less than 9.6.
- **Tented arch parameters** (x, y) **and** h: Along the y direction, we restrict the bottom of the arch (x, y) to lie in the bottom half of the thumbprint. We further estimate that x lies in the middle two-thirds of X. These assumptions yield $x \in (1.8, 9)$ and $y \in (0, 8.1)$. Assuming a symmetric distribution of (x, y) yields $(x, y) = (5.4, 2.8)$ with a standard deviation of 0.7 in both directions. Logically, we place h between 0 and $Y/2 = 8.1$. Again, assuming a symmetric distribution in this parameter space and a standard deviation of one-eighth the maximum value yields $h = 4.05 \pm 1.02$.

- **Whorl parameters** (x_L, y_L)**,** (x_C, y_C)**,** (x_R, y_R)**, and** (x_B, y_B): We expect (x_L, y_L) to be in the bottom left quadrant for all but the most extreme examples and similarly (x_R, y_R) to lie in the bottom right quadrant. We place (x_B, y_B) between $x = X/4$ and $x = 3X/4$ and $y = 0$ and $y = 2Y/3$. The topmost point, (x_C, y_C), we place in the top half of the thumbprint. We again put the average values in the center of their restricted areas.

Table 2 summarizes the estimates for these four classes of ridge structures.

Table 2. Parameter range estimates for the ridge structure classes.
All lengths in millimeters (mm), angles in degrees.

Arch parameter ranges	
(x, y)	$(4.2, 2.8) \pm (0.7, 0.7)$
h	4.05 ± 0.7
b	2.5 ± 0
θ_1–θ_4	$22.5° \pm 5.13°$
Loop parameter ranges	
(x, y)	$(2.7, 2.8) \pm (0.7, 0.7)$
θ	$45° \pm 15°$
r	4.58 ± 0.7
Tented Arch Parameter Ranges	
(x, y)	$(5.4, 2.8) \pm (0.7, 0.7)$
h	4.05 ± 1.02
Whorl parameter ranges	
(x_L, y_L)	$(2.7, 4.1) \pm (0.7, 0.7)$
(x_C, y_C)	$(5.4, 12.2) \pm (0.7, 0.7)$
(x_R, y_R)	$(8.1, 4.1) \pm (0.7, 0.7)$
(x_B, y_B)	$(5.4, 4.1) \pm (0.7, 0.7)$

Level-Two Parameters

- **Level-two spatial resolution limit** δ_2: We estimate δ_2 by r_0, the spatial uncertainty of minutiae location in two dimensions [Pankanti et al. 2002], and propose $\delta_2 = 1$ for best-case calculations.

- **Relative minutiae type frequencies** p_d**,** p_b**, and** p_t: Almost every compound minutia can be broken into a combination of bifurcations and terminations separated spatially. Counting these compound minutiae appropriately, we determine the relative minutiae frequencies in **Table 3**.

- **Ridge period** w: We use 0.463 mm / ridge for the ridge period, the distance from the middle of one ridge to the middle of an adjacent one [Stoney and Thornton 1986].

Table 3. Frequencies of simple minutiae types (from Osterburg et al. [1977]).

p_b	p_t	p_d
0.356	0.581	0.0629

- **Mean number of minutiae per print** μ: Under ideal circumstances, we discern 40 to 60 minutiae on a print [Pankanti et al. 2002]; we take $\mu = 50 \pm 10$.
- **Linear minutiae density** λ: We calculate λ by dividing the average number of minutiae per a thumbprint μ by the total ridge length of a thumbprint XY/w. Under ideal conditions, this gives $\lambda = 0.13 \pm 0.03$ minutiae / mm. In practice, we may have $\lambda = 0.05 \pm 0.03$ minutiae / mm [Pankanti et al. 2002].

Finally, we estimate that there have been 100 billion humans [Haub 1995].

Model Analysis and Testing

Let the probability that a print has a configuration x be $p_c(x)$. Assuming that fingerprint patterns are distributed independently, the probability that two prints match is $p_c^2(x)$. The sum of these probabilities over the configuration space is the total probability that some match occurs.

The probabilities associated with the two levels of detail are determined independently, so the total occurrence probability factors into $p_{c1}(x_1)p_{c2}(x_2)$. Denoting the level-one configuration subspace as C_1 and the level-two subspace as C_2, the total probability of the prints matching is

$$p = \sum_{i \in C_1} \sum_{j \in C_2} [p_{c1}(i) p_{c2}(j)]^2 = \left(\sum_{i \in C_1} p_{c1}^2(i) \right) \left(\sum_{j \in C_2} p_{c2}^2(j) \right) = p_1 p_2.$$

Level-One Detail Matching

We restrict each parameter to a region of parameter space in which we expect to find it and assume that it is uniformly distributed there. This approximation is enough to estimate order of magnitude, which suffices for our analysis. Then

$$p_{c1}(i) = \frac{\nu_i}{\left(\prod_{j \in V(i)} \dfrac{L_j}{\delta_1} \right)}, \qquad \textbf{(1)}$$

where L_j is the range of parameter j in $V(i)$, the set of parameters for a type-i ridge structure. For **(1)** to be accurate, we should make any L_j corresponding to angular parameters the product of the angle range with our typical length of 5.4 mm. The product is simply the total number of compartments in the parameter space, since we assume a uniform distribution in that range. Calculating $p_{c1}(i)$ for each ridge structure type, and summing squares, we find the probability that two thumbprints have the same overall ridge structure:

$$p_1 = \sum_{i \in C_1} p_{c1}^2(i) = .00044. \tag{2}$$

Level-Two Detail Matching

If we disregard the infrequent dot minutiae, we obtain the probability

$$p_{c2}(j) = (\delta_2\lambda)^k (1-\delta_2\lambda)^{C-K} p_b^{k_b} p_t^{k-k_b}\, \frac{1}{2^k}$$

for a configuration j with k minutiae, k_b of which are bifurcations (and the rest ridges), placed in $C = XY/w\delta_2$ cells. If we simplify minutia-type frequencies to $p_b = p_t = 1/2$, and note that there are

$$\binom{C}{k}\binom{k}{k_b}2^k$$

ways to configure j given k and k_b, the total probability of a match becomes

$$\begin{aligned}
p_2 &= \sum_{k=0}^{C}\sum_{k_b=0}^{k}\left((\delta_2\lambda)^k(1-\delta_2\lambda)^{C-K}\frac{1}{4^k}\right)\binom{C}{k}\binom{k}{k_b}2^k \\
&= \sum_{k=0}^{C}(\delta_2\lambda)^{2k}(1-\delta_2\lambda)^{2(C-k)}\frac{1}{4^k}\binom{C}{k} \\
&= \left(\frac{5(\delta_2\lambda)^2 - 8\delta_2\lambda + 4}{4}\right)^C.
\end{aligned}$$

Match probabilities for $\lambda = 0.13 \pm 0.03$/mm, $\delta_2 = 1$ mm, and $C = 250$ to 400 cells range from 2.9×10^{-23} to 9.8×10^{-60}; probabilities for the more realistic values $\lambda = 0.05 \pm 0.03$/mm, $\delta_2 = 2$–3 mm, and $C = 100$ to 250 cells range from 3.7×10^{-5} to 1.7×10^{-47}.

Historical Uniqueness of Fingerprints

Denote the probability of a match of any two left thumbprints in the history of the human race by p and the world total population by N. The probability of at least one match among $\binom{N}{2}$ thumbprints is

$$P = 1 - (1-p)^{\binom{N}{2}}.$$

Figure 4a illustrates the probability of at least one match for $N = 10^{11}$, while **Figure 4b** shows a log-log plot of the probability for very small p-values. Since even conservative parameter values in the ideal case give $p \ll 10^{-30}$, our model solidly establishes uniqueness of fingerprints through history.

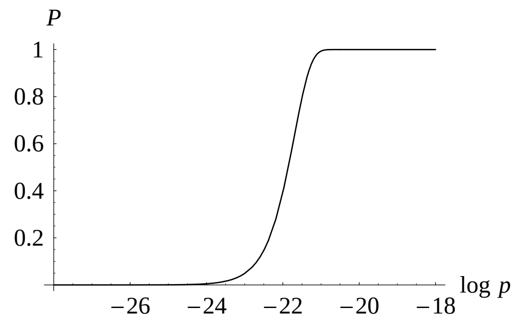


Figure 4a. For $N = 10^{11}$, probability of at least one thumbprint match through history.

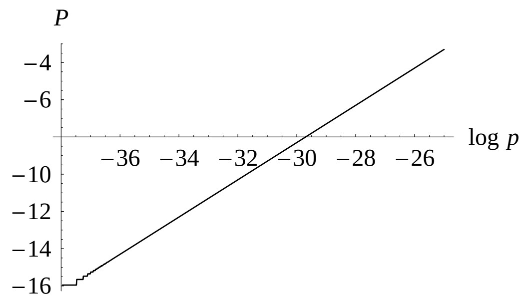


Figure 4b. Log-log plot of probability.

Strengths and Weaknesses of the Model

Strengths

- **Topological coordinate system**: We take topological considerations into account, as demanded by Stoney and Thornton [1986].
- **Incorporation of ridge structure detail**: We use this in addition to the minutiae detail that is the primary focus of most other models.
- **Integration of nonuniform distributions**: We allow for more-complex distributions of the ridge structure parameters, such as Gaussian distributions for singularity locations, and we consider that distribution of minutiae along ridges may depend on the location of the ridge in the overall structure.
- **Accurate representation of minutia type and orientation**: We follow models such as those developed by Roxburgh and Stoney in emphasizing the bidirectional orientation of minutiae along ridges, and we further consider the type of minutiae present as well as their location and orientation. Cruder models of minutiae structure [Osterburg et al. 1977; Pankanti et al. 2002] neglect some of this information.
- **Flexibility in parameter ranges**: We test a range of parameters in both ideal and practical scenarios and find that the model behaves as expected.

Weaknesses

- **Ambiguous prints, smearing, or partial matches**: We assume that ambiguities in prints reflect ambiguities in physical structure and are not introduced by the printing. This is certainly not the case for actual fingerprints.
- **Domain discontinuities**: We have no guarantee of continuity between regions of flow; continuity requirements may affect the level-one matching probabilities significantly.

- **Nonuniform minutia distribution**: We assume that the distribution of minutiae along a ridge is uniform. However, models should account for variations in minutiae density and clustering of minutiae [Stoney and Thornton 1986]. Although we have a mechanism for varying this distribution, we have no data on what the distribution should be.
- **Left/right orientation distribution**: We assume that the distribution of minutiae orientation is independent and uniform throughout the print. Amy notes, however, that the preferential divergence or convergence of ridges in a particular direction can lead to an excess of minutiae with a particular orientation [Stoney and Thornton 1986].
- **Level-three information**: We neglect level-three information, such as pores and edge shapes, because of uncertainty about its reproducibility in prints.

Comparison with DNA Methods

DNA Fingerprinting

The genetic material in living organisms consists of deoxyribonucleic acid (DNA), a macromolecule in the shape of a double helix with nitrogen-base “rungs” connecting the two helices. The configurations of these nitrogen bases encode the genetic information for each organism and are unique to the organism (except for identical twins and other cases in which an organism splits into multiple separate organisms).

Direct comparison of base-pair sequences for two individuals is infeasible, so scientists sequence patterns in a person’s DNA called *variable number tandem repeats (VNTR)*, sections of the genome with no apparent genetic function.

Comparison of Traditional and DNA Fingerprinting

While level-two data is often limited by print quality, we expect level-one detail to remain relatively unchanged unless significant sections of the print are obscured or absent. We use $p_1 = 10^{-3}$ from **(2)**, allowing for a conservative loss of seven-eighths of the level-one information. Multiplying by this level-one factor 10^{-3}, all but the three worst probabilities are less than 10^{-9}.

DNA fingerprinting has its flaws: False positives can arise from mishandling samples, but the frequency is difficult to estimate. The probability of two different patterns exhibiting the same VNTR by chance varies between 10^{-2} and 10^{-4}, depending on the VNTR [Roeder 1994; Woodworth 2001]. The total probability of an individual’s DNA pattern occurring by chance is computed under the assumption that the VNTRs are independent, which has been verified for the ten most commonly used VNTRs [Lambert et al. 1995].

Results and Conclusions

We present a model that determines whether fingerprints are unique. We consider both the topological structure of a fingerprint and the fine detail present in the individual ridges. We compute probabilities that suggest that fingerprints are reasonably unique among all humans who have lived.

Fingerprint evidence compares well with DNA evidence in forensic settings. Our model predicts that with even a reasonably small fingerprint area and number of features, the probability that a match between a latent print and a suspect's print occurs by chance is less than 10^{-9}. Both DNA evidence with few VNTRs and fingerprints of poor quality with few features can give inconclusive results, resulting in uncertainty beyond a reasonable doubt.

References

Adams, Julian. 2002. DNA fingerprinting, genetics and crime: DNA testing and the courtroom. `http://www.fathom.com/courese/21701758/` .

Beeton, Mary. 2002. Scientific methodology and the friction ridge identification process. *Identification Canada* 25 (3) (September 2002). Reprinted in *Georgia Forensic News* 32 (3) (November 2002) 1, 6–8. `http://www.ridgesandfurrows.homestead.com/files/Scientific_Methodology.pdf` .

Cappelli, Raffaele, Alessandra Lumini, Dario Maio, and Davide Maltoni. 1999. Fingerprint classification by directional image partitioning. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 21 (5): 402–421.

Epstein, Robert. 2002. Fingerprints meet *Daubert*: The myth of fingerprint "science" is revealed. *Southern California Law Review* 75: 605–658.

Galton, Francis. 1892. *Finger Prints*. London: Macmillan.

Haub, Carl. 1995. How many people have ever lived on Earth? *Population Today* 23 (2) (February) 4–5. Reprinted. 30 (8) (November / December 2002) 4–5. `http://www.prb.org/Content/ContentGroups/PTarticle/Oct-Dec02/How_Many_People_Have_Ever_Lived_on_Earth_.htm` .

Jeffreys, A.J., V. Wilson, and S.L. Thein. 1995. Individual-specific fingerprints of human DNA. *Nature* 316: 76–79.

Lambert, J.A., J.K. Scranage, and I.W. Evett. 1995. Large scale database experiments to assess the significance of matching DNA profiles. *International Journal of Legal Medicine* 108: 8–13.

Marcialis, Gian Luca, Fabio Roli, and Paolo Frasconi. 2001. Fingerprint classification by combination of flat and structural approaches. In *International*

Conference on Audio- and Video-Based Biometric Person Authentication AVBPA01 (June 2001, Halmstad, Sweden), edited by J. Bigun and F. Smeraldi. *Springer Lecture Notes in Computer Science* 2091: 241–246.

National Center for State Courts (NCSC). 2004. `http://www.ncsconline.org/` .

German, Ed. 2003. The history of fingerprints. `http://onin.com/fp/fphistory.html` .

Osterburg, James W., T. Parthasarathy, T.E.S. Ragvahan, and Stanley L. Sclove. 1977. Development of a mathematical formula for the calculation of fingerprint probabilities based on individual characteristics. *Journal of the American Statistical Association* 72: 772–778.

Pankanti, S., S. Prabhakar, and A. Jain. 2002. On the individuality of fingerprints. *IEEE Transaction on Pattern Analysis and Machine Intelligence*: 1010–1025. `http://citeseer.ist.psu.edu/472091.html` .

Prabhakar, Salil. 2001. Fingerprint matching and classification using a filterbank. Ph.D. thesis. East Lansing, Michigan: Michigan State University.

Roeder, K. 1994. DNA fingerprinting: a review of the controversy. *Statistical Science* 9: 222–247.

Sclove, Stanley L. 1979. The occurrence of fingerprint characteristics as a two-dimensional process. *Journal of the American Staistical Association* 74: 588–595.

Stoney, David A., and John I. Thornton. 1986. A critical analysis of quantitative fingerprint individuality models. *Journal of Forensic Sciences* 31 (4): 1187–1216.

Thompson, William. 2003. Examiner bias. `http://www.bioforensics.com/conference/ExaminerBias/` .

Watson, C.I., and C.L. Wilson. 1992. *NIST Special Database 4, Fingerprint Database*. U.S. National Institute of Standards and Technology.

Wayman, James L. 2000. Daubert hearing on fingerprinting: When bad science leads to good law: The disturbing irony of the *Daubert* hearing in the case of *U.S. v. Byron C. Mitchell*. `http://www.engr.sjsu.edu/biometrics/publications_daubert.html` .

Woodworth, George. 2001. Probability in court—DNA fingerprinting. `http://www.stat.uiowa.edu/~gwoodwor/statsoc` .

The Other Problems of the Second Ten Years

1995: The College Salaries Problem

Aluacha Balaclava College has just hired a new Provost. Problems with faculty compensation at the college forced the former Provost to resign, so the new Provost needs to make the institution of a fair and reasonable compensation system her first priority. As a first step in this process, she has hired your team as consultants to design a compensation system that reflects the following circumstances and principles.

Circumstances

There are four faculty ranks: Instructor, Assistant Professor, Associate Professor and Professor, in ascending order. Faculty with Ph.D. degrees are hired at the rank of Assistant Professor. Faculty who are working on a Ph.D. are hired at the rank of Instructor and promoted automatically to Assistant Professor upon completion of their degrees. Faculty may apply for promotion from Associate Professor to Professor after serving at the rank of Associate for seven or more years. The promotion decisions are made by the Provost with recommendations from a faculty committee and are not your concern.

Faculty salaries are for the ten-month period September through June. Raises are always effective beginning in September. The total amount of money available for raises varies from year to year and generally is not known until March for the following year.

The starting salary this year for an Instructor with no prior teaching experience was $27,000 and for an Assistant Professor was $32,000. Faculty can receive credit, upon hire, for as much as seven years of teaching experience at other institutions.

Principles

- All faculty should get a raise any year that money is available.
- Faculty should get a substantial benefit from promotion. If one is promoted in the minimum possible time, the benefit should be roughly equal to seven years of normal (non-promotion) raises.

Table 1.

Salary data for Problem B.

Case	Years	Rank	Salary	Case	Years	Rank	Salary	Case	Years	Rank	Salary
1	4	ASSO	54,000	2	19	ASST	43,508	3	20	ASST	39,072
4	11	PROF	53,900	5	15	PROF	44,206	6	17	ASST	37,538
7	23	PROF	48,844	8	10	ASST	32,841	9	7	ASSO	49,981
10	20	ASSO	42,549	11	18	ASSO	42,649	12	19	PROF	60,087
13	15	ASSO	38,002	14	4	ASST	30,000	15	34	PROF	60,576
16	28	ASST	44,562	17	9	ASST	30,893	18	22	ASSO	46,351
19	21	ASSO	50,979	20	20	ASST	48,000	21	4	ASST	32,500
22	14	ASSO	38,462	23	23	PROF	53,500	24	21	ASSO	42,488
25	20	ASSO	43,892	26	5	ASST	35,330	27	19	ASSO	41,147
28	15	ASST	34,040	29	18	PROF	48,944	30	7	ASST	30,128
31	5	ASST	35,330	32	6	ASSO	35,942	33	8	PROF	57,295
34	10	ASST	36,991	35	23	PROF	60,576	36	20	ASSO	48,926
37	9	PROF	57,956	38	32	ASSO	52,214	39	15	ASST	39,259
40	22	ASSO	43,672	41	6	INST	45,500	42	5	ASSO	52,262
43	5	ASSO	57,170	44	16	ASST	36,958	45	23	ASST	37,538
46	9	PROF	58,974	47	8	PROF	49,971	48	23	PROF	62,742
49	39	ASSO	52,058	50	4	INST	26,500	51	5	ASST	33,130
52	46	PROF	59,749	53	4	ASSO	37,954	54	19	PROF	45,833
55	6	ASSO	35,270	56	6	ASSO	43,037	57	20	PROF	59,755
58	21	PROF	57,797	59	4	ASSO	53,500	60	6	ASST	32,319
61	17	ASST	35,668	62	20	PROF	59,333	63	4	ASST	30,500
64	16	ASSO	41,352	65	15	PROF	43,264	66	20	PROF	50,935
67	6	ASST	45,365	68	6	ASSO	35,941	69	6	ASST	49,134
70	4	ASST	29,500	71	4	ASST	30,186	72	7	ASST	32,400
73	12	ASSO	44,501	74	2	ASST	31,900	75	1	ASSO	62,500
76	1	ASST	34,500	77	16	ASSO	40,637	78	4	ASSO	35,500
79	21	PROF	50,521	80	12	ASST	35,158	81	4	INST	28,500
82	16	PROF	46,930	83	24	PROF	55,811	84	6	ASST	30,128
85	16	PROF	46,090	86	5	ASST	28,570	87	19	PROF	44,612
88	17	ASST	36,313	89	6	ASST	33,479	90	14	ASSO	38,624
91	5	ASST	32,210	92	9	ASSO	48,500	93	4	ASST	35,150
94	25	PROF	50,583	95	23	PROF	60,800	96	17	ASST	38,464
97	4	ASST	39,500	98	3	ASST	52,000	99	24	PROF	56,922
100	2	PROF	78,500	101	20	PROF	52,345	102	9	ASST	35,798
103	24	ASST	43,925	104	6	ASSO	35,270	105	14	PROF	49,472
106	19	ASSO	42,215	107	12	ASST	40,427	108	10	ASST	37,021
109	18	ASSO	44,166	110	21	ASSO	46,157	111	8	ASST	32,500
112	19	ASSO	40,785	113	10	ASSO	38,698	114	5	ASST	31,170
115	1	INST	26,161	116	22	PROF	47,974	117	10	ASSO	37,793
118	7	ASST	38,117	119	26	PROF	62,370	120	20	ASSO	51,991
121	1	ASST	31,500	122	8	ASSO	35,941	123	14	ASSO	39,294
124	23	ASSO	51,991	125	1	ASST	30,000	126	15	ASST	34,638
127	20	ASSO	56,836	128	6	INST	35,451	129	10	ASST	32,756
130	14	ASST	32,922	131	12	ASSO	36,451	132	1	ASST	30,000
133	17	PROF	48,134	134	6	ASST	40,436	135	2	ASSO	54,500
136	4	ASSO	55,000	137	5	ASST	32,210	138	21	ASSO	43,160
139	2	ASST	32,000	140	7	ASST	36,300	141	9	ASSO	38,624
142	21	PROF	49,687	143	22	PROF	49,972	144	7	ASSO	46,155
145	12	ASST	37,159	146	9	ASST	32,500	147	3	ASST	31,500
148	13	INST	31,276	149	6	ASST	33,378	150	19	PROF	45,780
151	4	PROF	70,500	152	27	PROF	59,327	153	9	ASSO	37,954
154	5	ASSO	36,612	155	2	ASST	29,500	156	3	PROF	66,500
157	17	ASST	36,378	158	5	ASSO	46,770	159	22	ASST	42,772
160	6	ASST	31,160	161	17	ASST	39,072	162	20	ASST	42,970
163	2	PROF	85,500	164	20	ASST	49,302	165	21	ASSO	43,054
166	21	PROF	49,948	167	5	PROF	50,810	168	19	ASSO	51,378
169	18	ASSO	41,267	170	18	ASST	42,176	171	23	PROF	51,571
172	12	PROF	46,500	173	6	ASST	35,798	174	7	ASST	42,256
175	23	ASSO	46,351	176	22	PROF	48,280	177	3	ASST	55,500
178	15	ASSO	39,265	179	4	ASST	29,500	180	21	ASSO	48,359
181	23	PROF	48,844	182	1	ASST	31,000	183	6	ASST	32,923
184	2	INST	27,700	185	16	PROF	40,748	186	24	ASSO	44,715
187	9	ASSO	37,389	188	28	PROF	51,064	189	19	INST	34,265
190	22	PROF	49,756	191	19	ASST	36,958	192	16	ASST	34,550
193	22	PROF	50,576	194	5	ASST	32,210	195	2	ASST	28,500
196	12	ASSO	41,178	197	22	PROF	53,836	198	19	ASSO	43,519
199	4	ASST	32,000	200	18	ASSO	40,089	201	23	PROF	52,403
202	21	PROF	59,234	203	22	PROF	51,898	204	26	ASSO	47,047

- Faculty who get promoted on time (after seven or eight years in rank) and have careers of 25 or more years should make roughly twice as much at retirement as a new Ph.D. starting off.
- Faculty in the same rank with more experience should be paid more than others with less experience. But the effect of an additional year of experience should diminish over time. In other words, if two faculty stay in the same rank, their salaries should tend to get closer over time.

The Project

First, design a new pay system without cost-of-living increases. Then incorporate cost-of-living increases. The final piece of this project is to design a transition process for existing faculty that will move all salaries towards your system without cutting anyone's salary. The existing faculty salaries, ranks and years of service, are in **Table 1**. Discuss any refinements that you think would improve your system.

The Provost has asked for a detailed pay system plan that she can use for implementation, as well as a short executive summary in clear language, which she can present to the Board and to the faculty. The summary should outline the model, its assumptions, its strengths and weaknesses, and the expected results.

Comments

The Outstanding papers were by teams from Harvey Mudd College, North Carolina School of Science and Mathematics, Southeast Missouri State University, and University of Alaska Fairbanks. Their papers, together with commentaries, were published in *The UMAP Journal* 16 (3) (1995): 259–314.

Problem Origin

The problem was contributed by Kathleen M. Shannon (Salisbury State University, MD). The data were public information from Salisbury State University.

Judge's Comments

Contest judge Don Miller (St. Mary's College, IN) noted that most teams recognized the problem as a curve-fitting problem and found a model that satisfied most of the criteria.

Implementing the model on the provided data set turned out to be deceptively more difficult. The team from University Alaska Fairbanks had experimented with six different functions before settling on a logistic model because it met all the criteria.

Most teams used the same model for all ranks and the entire tenure of the faculty member. Others developed a separate model for the instructor rank or for the full professor rank (different because a full professor has no opportunity for promotion). The team from Southeast Missouri State University developed a separate model for each rank.

Modelers can gain creditability by demonstrating that they understand the problem in its context. For the salary data, this could mean recognizing some of the salaries as outliers that must be dealt with individually. The professor with two years of experience and a salary of $85,500 is clearly an outlier and will be virtually impossible to bring into line with any reasonable model. This person may be a "superstar" who is not expected to fit into the model salary structure (the football coach? no, the salary is too low). The modeler should point out or question such unusual situations, but few teams did. An exception was the team from the North Carolina School of Science and Mathematics, who showed that, "with the exception of a few grossly overpaid faculty, the problem of unfairness would be solved" in five years.

Better papers were distinguished by a complete and mature treatment of the assumptions as well as a precise plan of implementation. The team from University of Alaska at Fairbanks offered two plans, showed graphically how they differed, and recommended one over the other. All the Outstanding papers did sensitivity analyses, included cost of living in their implementation plan, and featured a well-written executive summary.

1996: The Submarine Detection Problem

The world's oceans contain an ambient noise field. Seismic disturbances, surface shipping, and marine mammals are sources that, in different frequency ranges, contribute to this field. We wish to consider how this ambient noise might be used to detect large moving objects, e.g., submarines located below the ocean surface. Assuming that a submarine makes no intrinsic noise, develop a method for detecting the presence of a moving submarine, its speed, its size, and its direction of travel, using only information obtained by measuring changes to the ambient noise field. Begin with noise at one fixed frequency and amplitude.

Comments

The Outstanding papers were by teams from Pomona College, University of North Carolina, Wake Forest University, and Worcester Polytechnic Institute. Their papers, together with commentaries, were published in *The UMAP Journal* 17 (3) (1996): 207–282.

Problem Origin

The problem was contributed by Daniel Zwillinger (Zwillinger & Associates).

Judge's Comments

Jack Robertson (then at Georgia College and State University, now at Georgia Military College) observed that it was critical for a team to consider the environmental effect of the ocean on sound propagation and account for the properties of the ambient noise field itself. Although the literature contains extensive discussions of both ideas, too many teams did little or nothing in this area. A great many papers made absolutely no attempt to do any true acoustic modeling. Instead, they looked like homework sets for a signal-processing course, rolling out page after page of theory without ever making a clear linkage to the problem posed. Those papers contained very little modeling—and modeling, after all, is what the contest is all about. Papers with simple models that were well conceived and whose shortcomings were clearly noted tended to fare much better than papers with extremely elaborate calculations and little connection with the real world.

Several papers stood out for creative schemes for designing receivers to work well under the conditions specified in the problem.

Many teams made little or no effort to search the literature. Very few teams referred to Buckingham et al. [1996], which appeared in a widely read science magazine shortly before the competition began.

Practitioner's Comments

Michael J. Buckingham (Scripps Institution of Oceanography and the University of Southampton) surveyed the history of underwater detection, including sonar, and the alternative based on the idea that ambient noise in the ocean acts as a form of acoustic illumination. An object will scatter and reflect some of the incident noise, suggesting that by focusing the scattered component with a suitable acoustic lens, it should be possible to create an image of the object. This, after all, is the essential process of photography using daylight.

The Worcester Polytechnic Institute paper is interesting in that it considered various types of reflector as candidates for an acoustic lens. In fact, a similar approach was implemented in the original investigations of ambient noise detection, in Buckingham et al. [1992] (a parabolic dish) and in Buckingham et al. [1996] (a larger spheroidal dish). This paper was the only one of the four to cite a reference to the research that has been performed on ambient noise detection and imaging. The authors proposed an interesting triangulation technique for parabolic reflectors and went on to consider two other detector geometries, a parabolic trough, and a parabolic torus. They found that the technique would work, but only over limited ranges, out to about 1 km, due to the absorption of sound by seawater.

The Wake Forest University team took a different approach by developing a computer simulation of noise detection with four close-packed hydrophones at each of four locations. They proposed a technique to give the submarine's position and, through a Doppler shift in the noise spectrum, its velocity. The idea of determining target speed from the noise field is novel, although probably would be difficult to achieve in practice, since the noise is broadband, whereas the authors assumed a single frequency. Another of the assumptions adopted by the authors of this paper is that the intensity of the noise is statistically stationary everywhere, which in reality is definitely not the case.

An extensive discussion of ambient noise, its properties, and sources is given by the Pomona College team. The authors proposed an array of directional hydrophones for detecting the presence of a submarine, plus a second array to give a three-dimensional view. This is the only paper that comes close to the idea that the ambient noise can be used not only for detection but also for imaging. They also went beyond theoretical analysis to perform simple experiments in air, using loudspeakers to simulate an ambient noise field and a rolling trash can as the target. To producing images, they propose a sound camera and also a variant of Schlieren photography, using a laser interference technique to observe the pressure fluctuations in the sound field. The authors were unfamiliar with Schlieren methods but worked out the essential principles for themselves.

The University of North Carolina team represented the submarine as an ellipsoidal object obstructing the noise field, giving rise to a reduction in noise intensity (a silhouette) at the receiver. Contouring of the noise intensity forms the basis of the recognition algorithm, with the velocity obtained by observing the submarine position at successive times.

The quality of the papers is very high, and some of the ideas presented may well work in practice. None of the teams considered using a multi-beam phased array as an acoustic lens. Such a system is a strong contender for future ambient noise imaging systems.

References

Buckingham, M.J. 1993. Theory of acoustic imaging in the ocean with ambient noise. *Journal of Computational Acoustics* 1: 117–140.

________, B.V. Berkhout, and S.A.L. Glegg. 1992. Imaging the ocean with ambient noise. *Nature* 356: 327–329.

________, John R. Potter, and Chad L. Epifanio. 1996. Seeing underwater with background noise. *Scientific American* 274 (2) (February 1996): 86–90.

Lasky, M. 1977. Review of undersea acoustics to 1950. *Journal of the Acoustical Society of America* 61: 283–297.

Potter, J.R. 1994. Acoustic imaging using ambient noise: Some theory and simulation results. *Journal of the Acoustical Society of America* 95: 21–33.

1997: The Velociraptor Location Problem

The velociraptor, *Velociraptor mongoliensis*, was a predatory dinosaur that lived during the late Cretaceous period, approximately 75 million years ago. Paleontologists think that it was a very tenacious hunter and may have hunted in pairs or larger packs. Unfortunately, there is no way to observe its hunting behavior in the wild, as can be done with modern mammalian predators. A group of paleontologists has approached your team and asked for help in modeling the hunting behavior of the velociraptor. They hope to compare your results with field data reported by biologists studying the behaviors of lions, tigers, and similar predatory animals.

The average adult velociraptor was 3 m long with a hip height of 0.5 m and an approximate mass of 45 kg. It is estimated that the animal could run extremely fast, at speeds of 60 km / h, for about 15 sec. After the initial burst of speed, the animal needed to stop and recover from a buildup of lactic acid in its muscles.

Suppose that velociraptor preyed on *Thescelosaurus neglectus*, a herbivorous biped approximately the same size as the velociraptor. A biomechanical analysis of a fossilized thescelosaurus indicates that it could run at a speed of about 50 km / h for long periods of time.

Part 1

Assuming the velociraptor is a solitary hunter, design a mathematical model that describes a hunting strategy for a single velociraptor stalking and chasing a single thescelosaurus as well as the evasive strategy of the prey. Assume that the thescelosaurus can always detect the velociraptor when it comes within 15 m, but may detect the predator at even greater ranges (up to 50 m) depending upon the habitat and weather conditions. Additionally, due to its physical structure and strength, the velociraptor has a limited turning radius when running at full speed. This radius is estimated to be three times the animal's hip height. On the other hand, the thescelosaurus is extremely agile and has a turning radius of 0.5 m.

Part 2

Assuming more realistically that the velociraptor hunted in pairs, design a new model that describes a hunting strategy for two velociraptors stalking and chasing a single thescelosaurus as well as the evasive strategy of the prey. Use the other assumptions and limitations given in Part 1.

Comments

The Outstanding papers were by teams from Calvin College, Harvard University, Pomona College, University of Alaska Fairbanks, and Washington University. Their papers, together with an author-judge's commentary, were published in *The UMAP Journal* 18 (3) (1997): 213–295.

Problem Origin

The problem was contributed by Jack Robertson (Mathematics Dept.) and William Wall (Dept. of Biological and Environmental Sciences), both of Georgia College and State University, Milledgeville, GA.

Author-Judge's Comments

Jack Robertson (now at Georgia Military College) commented that the data provided suggest unrealistically high forces when either animal made a sharp turn, a maneuver almost certainly necessary for survival. It was important to the success of the teams that they identified the data provided as containing an evident weakness and reacted in some reasonable and appropriate way. For example, quite a few teams drew on information readily available in the literature concerning mammalian species similar in size and behavior traits to the dinosaurs in the problem. This enabled the teams to adjust the data in a realistic way.

Another major issue in assumptions dealt with the geometry and mechanisms of the stalk, the chase, and the capture. The best teams provided clear, detailed thinking about their choices.

Some teams formulated models that used just algebra and geometry and no calculus. Others used differential equations, and one of the best papers used differential game theory.

This problem was especially well suited to graphical interpretation, and most teams provided graphs and charts that depicted the conduct and outcome of chases with one or two predators. Graphical analysis is particularly important when working with clients who may not grasp all the technical mathematics (such as paleontologists), and the illustrations produced by teams were absolutely vital.

1998: The Scanner Problem

Introduction

Industrial and medical diagnostic machines known as Magnetic Resonance Imagers (MRI) scan a three-dimensional object, such as a brain, and deliver their results in the form of a three-dimensional array of pixels. Each pixel consists of one number, indicating a color or a shade of gray that encodes a measure of water concentration in a small region of the scanned object at the location of the pixel. For instance, 0 can picture high water concentration in black (ventricles, blood vessels), 128 can picture a medium water concentration in gray (brain nuclei and gray matter), and 255 can picture a low water density in white (lipid-rich white matter consisting of myelinated axons). Such MRI scanners also include facilities to picture on a screen any horizontal or vertical slice through the three-dimensional array (slices are parallel to any of the three Cartesian coordinate axes).

Algorithms for picturing slices through oblique planes, however, are proprietary. Current algorithms

- are limited in terms of the angles and parameter options available,
- are implemented only on heavily used dedicated workstations,
- lack input capabilities for marking points in the picture before slicing, and
- tend to blur and "feather out" sharp boundaries between the original pixels.

A more faithful, flexible algorithm implemented on a personal computer would be useful

- for planning minimally invasive treatments;
- for calibrating the MRI machines;
- for investigating structures oriented obliquely in space, such as post-mortem tissue sections in animal research;
- for enabling cross sections at any angle through a brain atlas consisting of black-and-white line drawings.

To design such an algorithm, one can access the values and locations of the pixels but not the initial data gathered by the scanner.

Problem

Design and test an algorithm that produces sections of three-dimensional arrays by planes in any orientation in space, preserving the original gray-scale values as closely as possible.

Data Sets

The typical data set consists of a three-dimensional array A of numbers $A(i,j,k)$, where $A(i,j,k)$ is the density of the object at the location $(x,y,z)_{i,j,k}$. Typically, $A(i,j,k)$ can range from 0 through 255. In most applications, the data set is quite large. Teams should design data sets to test and demonstrate their algorithms. The data sets should reflect conditions likely to be of diagnostic interest. Teams should also characterize data sets that limit the effectiveness of their algorithms.

Summary

The algorithm must produce a picture of the slice of the three-dimensional array by a plane in space. The plane can have any orientation and any location in space. (The plane can miss some or all data points.) The result of the algorithm should be a model of the density of the scanned object over the selected plane.

Comments

The Outstanding papers were by teams from Eastern Oregon University, Harvey Mudd College, Macalester College, and Tsinghua University. Their papers, together with commentaries, were published in *The UMAP Journal* 19 (3) (1998): 211–278.

Problem Origin

The problem was contributed by Yves Nievergelt (Eastern Washington University).

Author's Comments

The problem came from the laboratory of Dr. Mark F. Dubach in studying the effects of intracerebral drug injections on monkeys with brain diseases at the University of Washington's Regional Primate Research Center in Seattle, WA.

Striking about the Outstanding solutions was the teams' mastery of finding information on the World Wide Web, their use of computer graphics (no numerical summary can substitute for a final visual medical diagnostic), and their computer programming (for the change of coordinates, in effect an isometric parametrization of a plane in space, and for three-dimensional interpolation).

All of the Outstanding teams realized that the practical problem could be cast as a mathematical problem in three-dimensional interpolation. Detail was important in the generalization from one- or two-dimensional to three-dimensional interpolation. While one team (Tsinghua University) already knew the result, other teams (Eastern Oregon University, Harvey Mudd College) offered excellent explanations and proofs of their mathematical generalizations.

The teams demonstrated efficient time tradeoff between searches and in-house production of data and algorithms. Such a balancing act between finding and reinventing the wheel can be critical to deliver a report on time. None of the teams used a three-dimensional interpolation computer program from the World Wide Web, perhaps because it is not obvious where to get one. Such routines exist, but finding them and using them may demand more time than available. For instance, there is a multidimensional (with an unlimited number of dimensions) interpolation routine using nonuniform rational B-splines (NURBS) at `http://dtnet33-199.dt.navy.mil/dtnurbs/about.htm`.

Judge's Comments

William P. Fox (Francis Marion University) noted that the judges did not witness a wide range of mathematical modeling; most teams recognized this problem as an image-processing problem.

One component of the problem required an oblique slice from an image scanned vertically or horizontally. Teams used one of three basic methods:

- (most frequent) Create a plane, $Ax + By + Cz = D$, and then rotate it using a standard matrix transformation.
- Select two points in three-space and define a plane between those points.
- Select one point and two angles to define the plane.

A critical element was mapping the coordinates of the oblique plane through the three-dimensional data set to obtain a gray scale scheme (0–255) for the elements in the plane. The three-dimensional data set was defined by three integers, while the points in the oblique plane were real numbers. Methods had to be developed to interpolate the grayscale values for all the points in the oblique plane:

- a nearest-neighbor algorithm, using eight or more points;
- a weighted-point algorithm;
- splines (linear through cubic); and
- Lagrangian polynomials.

Teams usually tried more than one method, though comparisons were generally sketchy and lacked analysis. Teams that critically compared and analyzed their methodology and results, and reached valid conclusions, impressed the judges.

The problem statement also required design and testing of an algorithm to produce sections of three-dimensional arrays made by planes in any orientation in space, preserving as closely as possible the original grayscale values. Some teams used color to enhance their presentation; this was acceptable provided that the grayscale was not totally removed. Several teams suggested color because their grayscale resolution could not detect certain diagnostic elements; this was viewed as a fatal flaw, since the problem statement required grayscale.

The team's algorithm had to produce a picture of the slice of the three-dimensional array by a plane in space. Judges wanted to see a picture, not a matrix portrayal. They scrutinized pictures closely to see if they appeared to be oblique slices.

The teams were expected to demonstrate their algorithm by designing and testing data sets that reflect conditions likely to be of diagnostic value. Teams should also characterize data sets that limit the effectiveness of their algorithm; few teams did so. Thus, judges looked for a good descriptions of the data sets and the elements of diagnostic value. Some teams created spheres inside cubes as their representative data set; provided teams put something of diagnostic value inside their lager 3-D elements, their data sets were acceptable.

Another important element not uniformly accomplished was some kind of error analysis. Very few teams even checked for accuracy their integer values in the plane against the corresponding integer point in their three-space data set. The judges praised teams that accomplished that.

1999: The Lawful Capacity Problem

Many facilities for public gatherings have signs that state that it is "unlawful" for their rooms to be occupied by more than a specified number of people. Presumably, this number is based on the speed with which people in the room could be evacuated from the room's exits in case of an emergency. Similarly, elevators and other facilities often have "maximum capacities" posted.

Develop a mathematical model for deciding what number to post on such a sign as being the "lawful capacity." As part of your solution, discuss criteria—other than public safety in the case of a fire or other emergency—that might govern the number of people considered "unlawful" to occupy the room (or space). Also, for the model that you construct, consider the differences between a room with movable furniture such as a cafeteria (with tables and chairs), a gymnasium, a public swimming pool, and a lecture hall with a pattern of rows and aisles. You may wish to compare and contrast what might be done for a variety of different environments: elevator, lecture hall, swimming pool, cafeteria, or gymnasium. Gatherings such as rock concerts and soccer tournaments may present special conditions.

Apply your model to one or more public facilities at your institution (or neighboring town). Compare your results with the stated capacity, if one is posted. If used, your model is likely to be challenged by parties with interests in increasing the capacity. Write an article for the local newspaper defending your analysis.

Comments

The Outstanding papers were by teams from Eastern Oregon University, Harvey Mudd College, Macalester College, and Tsinghua University. Their papers, together with commentaries, were published in *The UMAP Journal* 20 (3) (1999): 273–372.

Problem Origin

The problem was contributed by Joe Malkevitch (York College, City University of New York).

Practitioner's Comments

Richard Hewitt (US WEST Communications) commented from experience:

My hands-on training in public safety includes spending time in burning buildings, cutting people out of wrecked cars, and, unfortunately, putting a few people in body bags. As a result of these experiences, I developed sufficient knowledge and credibility to develop a method that the Denver Fire Department uses for selecting sites for new fire stations. The station-siting problem is similar to the contest problem in that my mathematical training enabled me to find the correct answer, but focusing exclusively on a mathematical solution moved me further away from a solution that could be implemented.

Common Assumptions were:

- "People will exit via the nearest exit." Unfortunately, this not the case. According to Denver Fire Chief Richard Gonzales, "in the case of fire, people exit via the door they entered, regardless of the nearest exit. . . . They generally remember the way they came in and retrace that path even if another exit is much closer." This finding further complicates the maximum occupancy problem and creates a need to understand how people enter a room or building to determine how they will exit. This adds a level of complexity to determining how quickly people will exit and therefore the maximum number to let in.

- "The average person can exit at a rate of x feet per second." I have been in several burning buildings where the smoke was so thick that I couldn't see my hand in front of my face. It's also not unusual for the power to go out because of the fire or water used to fight it. Unless the room has emergency lighting (or it's daytime and the room has exterior windows), you're moving around in darkness. In smaller fires, or in the early stages of a fire, it can be difficult to see across a well-lit room; and many public places, such as restaurants, bars, dance halls, and theaters, are not well lit. Visibility is a critical variable that impacts how fast people can exit or even find the exit that they remember entering.

- "People behave rationally in an emergency." In a private conversation, Chief Gonzales cited several examples of just how irrationally some people behave, such as the restaurant patron who refused to leave a burning building, even as the room was filling with smoke. The man argued with fire department personnel, "I paid for this steak and I'm going to eat it." This may be an extreme example, but it highlights a point: You have to account for human behavior, whether it's logical or not, because that behavior represents reality. Failure to do so leaves your model, and therefore your solution, open to attack.

To maximize the probability of a successful implementation, your solution, model, or method must address the issues of each stakeholder. So you must first figure out who the stakeholders are. In the case of the maximum occupancy problem, Chubb and Williamson [1998] provide a fairly complete list of construction project stakeholders, each of which has a stake in the maximum occupancy decision: owners, designers, builders, insurers, regulators, and occupants.

You have to identify the needs and concerns of each stakeholder group. This is best done by talking to them, in person, on their turf and in their terms. During the discussion, ask lots of "why" questions ("Why is that important?" and "Why do you feel that way?"). The answers enable you to understand what each stakeholder values. You can then define a solution space that incorporates what the stakeholders told you was important. You then begin to weigh priorities and make tradeoffs based on politics, economic impact, risk, importance to the decision makers, and / or the ability of a stakeholder group to block the implementation of your solution. And yes, you can now incorporate your mathematical findings.

I have never (outside of academia) seen a mathematical solution dominate the other decision criteria. In really good solutions, the mathematical findings complement the solution, but they never dictate it.

A newspaper article defending your method should focus on its ease of use, grounding in common sense, and the amount by which the results you generate exceed what they already have. A quote from a highly visible and respected official never hurts either.

As an example of what not to do, I submit the following.

> Dear Mr. And Mrs. Public,
>
> Concerning our award-winning method to determine the maximum occupancy level of your child's elementary school classroom, we used a polyhedral approach to approximate a statistically unbiased estimator that incorporated Euler's formula to model crowd movement based in small rooms. This model incorporated Chebyshev's inequality as it applies to elementary-school traffic patterns.
>
> We then fed our results into a simulation model utilizing software that we built ourselves based on tools that we downloaded from the Internet.
>
> While this method is probably way over your head, we have full confidence in its ability to forecast the probability of an emergency during school hours.
>
> This information enables us to set the maximum occupancy of your child's classroom at 183 plus or minus 7%.
>
> Yours truly,
> Contest Winners

The correct approach would be to convince the public that your method yields a solution that increases their safety and improves the likelihood of the survival of their loved ones.

Reference

Chubb, M.D., and R.B. Williamson. 1998. Value-based fire safety: A new regulatory model for mitigating human error. In *Human Behaviour In Fire—Proceeding of the First International Symposium*, August 30—September 2, 1998, edited by T.J. Shields, 105–114. Belfast, Northern Ireland: University of Ulster.

Judge's Comments

Jerry Griggs (University of South Carolina) noted that many teams took an overly simplified approach:

Determine

- an exit flow rate of r people per second per exit,
- the number of exits n in the room, and
- the number of seconds s to clear the room safely,

then obtain a room capacity of rns people.

However, the best papers allowed for a range of significant factors to be included in the model. Among these, they consider the flow of people through the room—not just at the exit—as well as crowd congestion due to bottlenecks created by the room shape and furniture placement. A strong model allows a variable initial distribution of people in the room.

Many entries omitted several of the elements requested in the problem:

- considering different room arrangements and environments,
- comparing models to posted requirements or codes,
- discussing criteria other than safety, and
- writing an explanatory article suitable for the newspaper.

Researching the problem impressed the judges, such as by consulting existing codes or by gathering data directly from crowd observations. One paper even considered capacity reductions mandated by the Americans with Disabilities Act!

Some papers present easily understood graphs of exit time as a function of the number of people in the room; such displays make it easy for the decision-maker.

It was nice to see analysis of model run-time complexity, which is important in dealing with very large or complicated arrangements, along with improvements made by simplifying calculations.

Papers stood out that consider factors that could be included in a more elaborate model, such as crowd panic, accessibility of emergency personnel, ventilation, and crowd flow out of the entire building.

One nice approach was to use a graphical / network flow model. One paper employs a series of queues to handle the bottlenecks. Another model tiles the room with one-person-sized hexagons and calculates the expected waiting times for each. There is a sophisticated motion simulation model that represents people by disks that naturally flow around obstacles towards exits.

2000: The Channel Assignment Problem

We seek to model the assignment of radio channels to a symmetric network of transmitter locations over a large planar area, so as to avoid interference. One basic approach is to partition the region into regular hexagons in a grid (honeycomb-style), as shown in **Figure 1**, where a transmitter is located at the center of each hexagon.

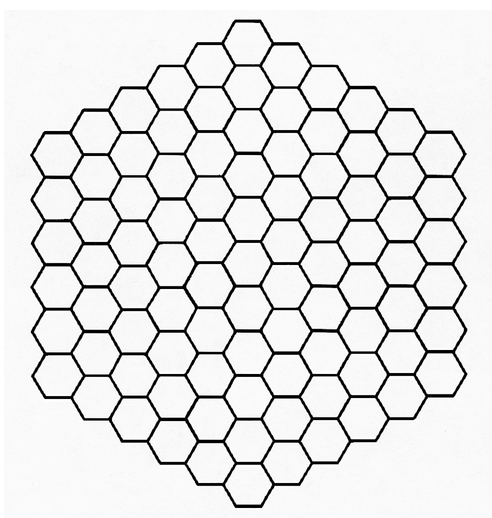


Figure 1. Honeycomb grid of hexagons.

An interval of the frequency spectrum is to be allotted for transmitter frequencies. The interval will be divided into regularly spaced channels, which we represent by integers 1, 2, 3, ... Each transmitter will be assigned one positive integer channel. The same channel can be used at many locations, provided that interference from nearby transmitters is avoided.

Our goal is to minimize the width of the interval in the frequency spectrum that is needed to assign channels subject to some constraints. This is achieved with the concept of a span. The span is the minimum, over all assignments satisfying the constraints, of the largest channel used at any location. It is not required that every channel smaller than the span be used in an assignment that attains the span.

Let s be the length of a side of one of the hexagons. We concentrate on the case that there are two levels of interference.

- Requirement A: There are several constraints on frequency assignments:
  - No two transmitters within distance $4s$ of each other can be given the same channel.
  - Due to spectral spreading, transmitters within distance $2s$ of each other must not be given the same or adjacent channels: Their channels must differ by at least 2.

  Under these constraints, what can we say about the span in Figure 1?
- Requirement B: Repeat Requirement A, assuming the grid in the example spreads arbitrarily far in all directions.
- Requirement C: Repeat Requirements A and B, except assume now more generally that channels for transmitters within distance $2s$ differ by at least some given integer k, while those at distance at most $4s$ must still differ by at least one. What can we say about the span and about efficient strategies for designing assignments, as a function of k?
- Requirement D: Consider generalizations of the problem, such as several levels of interference or irregular transmitter placements. What other factors may be important to consider?
- Requirement E: Write an article (no more than 2 pages) for the local newspaper explaining your findings.

Comments

The Outstanding papers were by teams from California Polytechnic State University, Lewis & Clark College, National University of Defence Technology (Changsha, China), Wake Forest University, and Washington University. Their papers, together with an author-judge's commentary, were published in *The UMAP Journal* 20 (3) (2000): 311–388.

Problem Origin

The problem was contributed by Jerrold R. Griggs (University of South Carolina).

Judge's Comments

The original "channel assignment problem" has a long history and detailed some of it. The problem is to assign an integer channel to each transmitter in a network, with the condition that the absolute difference between channels for two nearby transmitters must not belong to a certain set T that arises from interference considerations (see Hale [1980] for motivation). A feasible assignment can be obtained with channels far apart, but this is highly inefficient. Typically, a frequency band that spans the assigned channels is allocated to the network; the wider the band, the more it costs. The problem, then, is to minimize the "span" of the assignment, which is the difference between the maximum channel and the minimum channel.

This problem is modeled nicely with graph theory by letting each transmitter correspond to a vertex, with edges corresponding to pairs of nearby transmitters. The problem becomes a special vertex-coloring problem, owing to the set T of forbidden differences [Cozzens and Roberts 1982]. Among the methods that come into play are number theory (in the case of complete graphs [Griggs and Liu 1994]) and the complexity of graph homomorphisms.

The potential application is mobile radio networks. Large areas are often covered by a network of regularly spaced transmitters such that the associated graph labeling problem exactly models the network problem. The most common design places the transmitters in a triangular lattice.

For Requirement A, feasible solutions are easy to find, but working down to an optimal one requires some cleverness. Most teams achieved the minimum span.

Part B extends the network of Part A to the whole plane. While one can solve Part A by trial-and-error or by a computer search, Part B requires a method to keep going forever. Successful teams for Part B usually found a pattern (a strip or a tile of numbers) that could be repeated indefinitely and achieve the same span as the bounded array in Part A. One way is to label a strip by an appropriate ordering, say

$$1\ 3\ 5\ 7\ 9\ 2\ 4\ 6\ 8\ 1\ 3\ \ldots,$$

and then use the same strip shifted appropriately for the next row, and so on. Another perspective is to construct an appropriate tile of nine hexagons and replicate it. Judges were pleased to see papers that tested a variety of heuristics to assign channels in Parts A and B, since such methods are needed for more general arrays and distance parameters. At least one paper, by the team from Wake Forest University, made the interesting observation (with proof) that the optimal labeling for Parts A and B is essentially unique!

What is most remarkable is that several teams were able to solve Part C, in which the channel spread parameters for Parts A and B are extended to $d_1 = k, d_2 = 1$. It seems to be a new result.

Part D is the open-ended generalization of the problem to general array configurations and multiple levels of interference. It has the most room for creativity and for model design, but most entries did not do much here.

For Part E, judges wanted to see an article that conveys to the public the sense of the problem and the team's ideas on how to attack it. A particularly amusing article was crafted by a team from Harvey Mudd College whose entry received Honorable Mention.

Several teams located related results in the literature or the Web, particularly for the problem of *cyclic* labelings, where integers $\{1, 2, \ldots, n\}$ are used but the distance between two labels is measured modulo n, that is, by the shortest path on the circle labelled 1 through n. This approach can be used when a large number of channels must be assigned to each location: When a vertex receives label i, it is given all channels congruent to i modulo n. For two levels of interference ($L(d_1, d_2)$), this cyclic problem is solved in van den Heuvel et al. [1998]. However, this does not immediately solve the contest problem. A solution for general $L(d_1, d_2)$ of the (linear) contest problem remains to be found.

Teams typically found good labelings for the bounded array in Part A by trial and error or by exhaustive computer search, for small values of k, and identified patterns or tiles that could be extended to general k to yield good labelings for the bounded and unbounded arrays.

One cannot be certain that a labeling is optimal without proving that there is no labeling of smaller span. Also, it is by no means clear that there exist optimal labelings using a repeating pattern, although many teams seemed to assume this. Thus, it is not sufficient to check just labelings from a repeating pattern. (In fact, it would be very interesting if one could show that for all sets of distance parameters d_i that there is an optimal labeling of the plane built from a repeating pattern. This seems to be an open question.)

Judges favored papers that provide a *clear proof* that their labelings are optimal for general k. The best proofs that we read were impressive, such as the one by the team from the National University of Defence Technology. That paper is among those that made the interesting observation that for general k there is an optimal labeling for the arrays in Parts A and B that uses only nine different channels—which could be useful in some applications!

[EDITOR'S NOTE: Prof. Griggs's original commentary contains much more detail and cites many more references.]

References

Cozzens, M.B., and F.S. Roberts. 1982. T-colorings of graphs and the channel assignment problem. *Congressus Numerantium* 35: 191–208.

Griggs, Jerrold R., and D. D.-F. Liu. 1994. The channel assignment problem for mutually adjacent sites. *Journal of Combinatorial Theory A* 68: 169–183.

Hale, W.K. 1980. Frequency assignment: Theory and applications. *Proceedings of the IEEE* 68: 1497–1514.

van den Heuvel, J., R.A. Leese, and M.A. Shepherd. 1998. Graph-labeling and radio channel assignment. *Journal of Graph Theory* 29: 263–283.

2001: The Bicycle Wheel Problem

Introduction

Cyclists have different types of wheels they can use on their bicycles. The two basic types of wheels are those constructed using wire spokes and those constructed of a solid disk (see **Figure 2**). The spoked wheels are lighter but the solid wheels are more aerodynamic. A solid wheel is never used on the front for a road race but can be used on the rear of the bike.

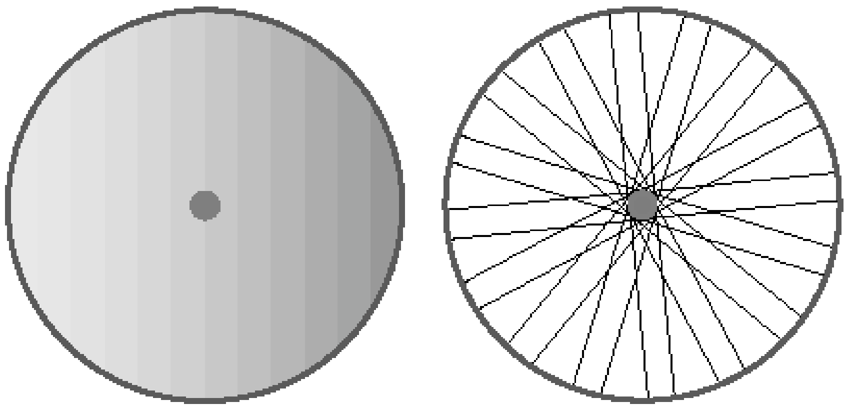


Figure 2. Solid wheel (left) and spoked wheel (right).

Professional cyclists look at a racecourse and make an educated guess as to what kind of wheels should be used. The decision is based on the number and steepness of the hills, the weather, wind speed, the competition, and other considerations.

The *directeur sportif* of your favorite team would like to have a better system in place and has asked your team for information to help determine what kind of wheel should be used for a given course.

The *directeur sportif* needs specific information to help make a decision and has asked your team to accomplish the tasks listed below. For each of the tasks, assume that the same spoked wheel will always be used on the front but that there is a choice of wheels for the rear.

Task 1

Provide a table giving the wind speed at which the power required for a solid rear wheel is less than for a spoked rear wheel. The table should include the wind speeds for different road grades starting from 0% to 10% in 1% increments. (Road grade is defined to be the ratio of the total rise of a hill divided by the length of the road.[1]) A rider starts at the bottom of the hill at a speed of 45 km / h

[1] If the hill is viewed as a triangle, the grade is the sine of the angle at the bottom of the hill.

and the deceleration of the rider is proportional to the road grade. A rider will lose about 8 km / h for a 5% grade over 100 m.

Task 2

Provide an example of how the table could be used for a specific time trial course.

Task 3

Determine if the table is an adequate means for deciding on the wheel configuration and offer other suggestions as to how to make this decision.

Comments

The Outstanding papers were by teams from Stellenbosch University (South Africa), U.S. Military Academy, and University College Cork (Ireland). Their papers, together with an author-judge's commentary, were published in *The UMAP Journal* 22 (3) (2001): 211–256.

Problem Origin

The problem was contributed by Kelly Black (University of New Hampshire).

Author-Judge's Comments

The problem focused on the two most basic types of wheels, the spoked wheel and the solid wheel. Spoked wheels have the lowest mass but the highest friction forces, due to interactions with air. Solid wheels have the most mass but the lowest friction forces. Three tasks were given:

- Find the wind speeds for which one wheel has an advantage over the other for particular inclines.
- Demonstrate how to use the information in the first task to determine which wheel to use for a specific course.
- Evaluate whether the information provided in the first task achieved the overall goal.

  A number of submissions concentrated on the *yaw angle*, the angle that the wind makes with respect to the direction of movement of the bicycle. While some of these submissions were quite good, others spent so much time trying to figure out how to deal with this complicated aspect that they did not make sufficient progress on other parts of the problem.

There were a number of entries in which Newton's Second Law, the torque equations, or the power was not correctly identified. There was also some confusion about units.

Overall, there were two different approaches:

- The first approach focused on the mechanics of a bicycle moving on an incline.

  The main difficulty was in isolating and identifying the relevant force based on Newton's Second Law and the corresponding torque equations. In many cases, it was difficult to identify exactly how the system of equations was manipulated and how the equations were found. The highly rated submissions did an excellent job of displaying and referring to the free-body diagrams, as well as discussing how the relevant force was isolated.

- The second approach focussed on the aerodynamic forces acting on the wheels, then calculated the power to move the wheels forward. For the spoked wheel, the total force acting on the wheel was found by adding the effects on each spoke (along its entire length) with its respective orientations. For the solid wheel, the forces acting on the whole wheel were found with respect to the wind yaw angle.

  This approach turned out to be difficult. The teams that carefully structured their approach and clearly identified each step stood out.

For either approach, it was common to make some assumption about either the acceleration or the steady-state velocity as the bike and rider moved along the hill. The second most common assumption was to assume that the rider provides a constant power output and then work backwards to isolate the forces acting on the wheels. The judges did not question the technical merits of these assumptions but concentrated instead on whether or not the submissions presented a clear and consistent case based on them.

The first task was specific and straightforward. How the two remaining tasks (use the table in a time trial, determine if the table is adequate) were addressed and presented was what set papers apart. The majority of submissions divided the race course into discrete pieces; they found the total power by adding up the power requirements over each piece.

Entries that impressed the judges went further. A few entries approached the third task by noting that the real goal was to minimize the time spent on a particular race course. By assuming that the rider would expend constant power output, the equations of motions from Newton's Second Law could then be found. The position of the rider on the course at any given time could then be approximated through a numerical integration of the resulting system of equations. For a given racecourse, the total time on the course could be found for different wheel configurations. A simple comparison of total times determined which wheel to use for the course.

2002: The Wind and Waterspray Problem

An ornamental fountain in a large open plaza surrounded by buildings squirts water high into the air. On gusty days, the wind blows spray from the fountain onto passersby. The water flow from the fountain is controlled by a mechanism linked to an anemometer (which measures wind speed and direction) located on top of an adjacent building. The objective of this control is to provide passersby with an acceptable balance between an attractive spectacle and a soaking: The harder the wind blows, the lower the water volume and the height to which the water is squirted, hence the less spray falls outside the pool area.

Your task is to devise an algorithm that uses data provided by the anemometer to adjust the water-flow from the fountain as the wind conditions change.

Comments

The Outstanding papers were by teams from Maggie L. Walker Governor's School, North Carolina School of Science and Mathematics, U.S. Military Academy, and University of Washington. Their papers, together with a judge's commentary, were published in *The UMAP Journal* 23 (3) (2002): 209–271.

Problem Origin

The problem was contributed by Tjalling Ypma (Western Washington University).

Judge's Comments

Patrick Driscoll (U.S. Military Academy) noted that the input quantities (anemometer data) and output quantities (water flow characteristics) for the algorithm required by the problem were clear from the problem description. The particular transformation rules for this problem were unspecified and left to the team to decide upon.

The predominant approach was Newton's Second Law of Motion, Bernoulli's formula, continuity equations, fuzzy membership sets, Poiseuille's equation, or the Navier-Stokes equations, largely dependent on the assumptions.

At least three categories of computational testing come to mind that support model *validation*:

- Once the team is convinced that their base model produces reasonable results, special cases of interest (e.g., no wind, no spread angle, etc.) should be tested.

- Recognizing that model parameters contain some amount of uncertainty, high, most likely, and low values of important parameters used in the base model should be examined by systematically altering these values and rerunning the model to see if the output results remain reasonable. For this MCM problem, these parameters might be drag coefficients, shapes of water droplets, wind speed and direction, and so on. This process essentially constitutes what is commonly referred to as *sensitivity analysis* of the parameters.

- The effects of relaxing a select number of simplifying assumptions made during the course of developing the model should be examined. However, it is fair to stress that this last category is safely performed only when time permits, because it generally requires substantial model modifications to examine the desired effects. A good example of this third category for the Wind and Waterspray problem would be adding the influence of surrounding buildings on wind speed and direction after they were previously assumed away. Such a change would be nontrivial and might consume more time than what is available.

Teams must link their computational results back to the problem that they are trying to solve. Tell the reader what to conclude from the results! This is what is referred to as *analyzing the results*.

2003: The Stunt Person Problem

An exciting action scene in a movie is going to be filmed, and you are the stunt coordinator! A stunt person on a motorcycle will jump over an elephant and land in a pile of cardboard boxes to cushion their fall. You need to protect the stunt person, and also use relatively few cardboard boxes (lower cost, not seen by camera, etc.).

Your job is to:

- determine what size boxes to use,
- determine how many boxes to use,
- determine how the boxes will be stacked,
- determine if any modifications to the boxes would help, and
- generalize to different combined weights (stunt person and motorcycle) and different jump heights.

Note that in the 1997 film "Tomorrow Never Dies," the James Bond character, on a motorcycle, jumps over a helicopter.

Comments

The Outstanding papers were by teams from California Institute of Technology, Harvard University, Harvey Mudd College, Southeast University (Nanjing, China), University of Washington, and Zhejiang University (Hangzhou, China). Their papers, together with a judge's commentary, were published in *The UMAP Journal* 24 (3) (2003): 221–324.

Problem Origin

The problem was contributed by Daniel Zwillinger (Zwillinger and Associates).

Judge's Comments

William P. Fox (Francis Marion University) enjoyed judging the problem because the teams did not have a wealth of information for the overall model from the Web or from other resources. Students could find basic information for helping model the jumping of the elephant from many sources. This problem turned out to be a "true" modeling problem; the students' assumptions led to the development of their model.

Six papers were selected as Outstanding submissions because they:

- developed a workable, realistic model from their assumptions and used it to answer all elements of the stunt person scenario;
- made clear recommendations as to the number of boxes used, their size, and how they should be stacked, and offered a generalization to other stunt persons;
- wrote a clear and understandable paper describing the problem, their model, and results; and
- handled all the elements.

The required elements, as viewed by the judges, were in two distinct phases.

- Models needed to consider the mission of the stunt person. A model had to be developed that ensured that the stunt person could jump over the elephant. The better teams then worked to minimize the speed with which to accomplish this jump. Teams that used a high-speed jump were usually discarded by the judges quickly.
- In Phase II, the model had to consider the landing; this included speed, energy, force, and momentum of the jumper, so that the boxes could be used safely to cushion the fall.

Most of the better papers did an extensive literature and Web search for information about cardboard. However, many teams spent way too much energy researching cardboard; their time would have been better spent in modeling.

The poorest section in all papers, including many of the Outstanding papers, was the summary.

Another flaw was the misuse of ECT (Edge Compression Testing) in a proportionality model. Many teams started with a Website formula,

$$BCS = 5.87 \times \text{ECT} \times \sqrt{Pt},$$

where P is the perimeter of the box and t is its thickness. They developed an energy model, where energy is the area under a curve, namely

$$\tfrac{1}{2} \times P \times \text{ECT} \times h$$

—a formula also at the Website—and reached conclusions based on using h as the height of the stacked boxes. Dan Zwillinger (the problem author) discovered both the use of the formula in many papers and the flaw in it. The potential energy is an area but it is

$$\tfrac{1}{2} \times P \times \text{ECT} \times \delta,$$

where δ is the depth of crush or deformation of the boxes, not the total height of the boxes (see **Figure 3**).

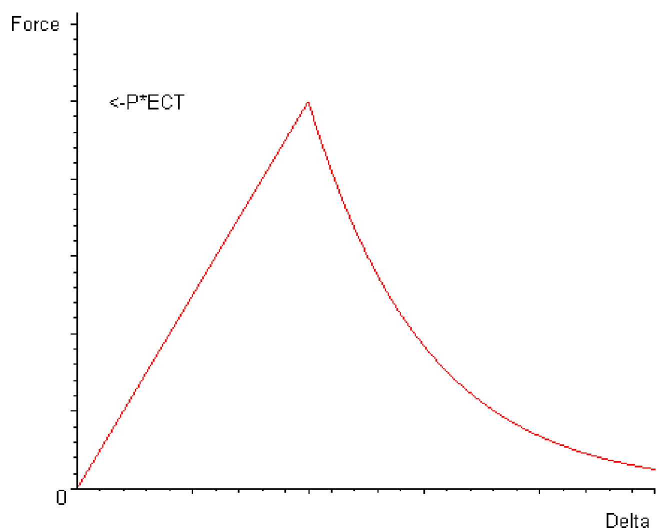


Figure 3. Force as a function of delta in ECT.

The moral, of course, is not to incorporate elements into your model that you don't understand thoroughly.

The better papers used a variety of methods to model the safe landing. These included kinematics, work, and energy absorption. The discussion of the boxes and how they were to be secured was also an important feature. One team laid many large mattress boxes flat on top of the stacked boxes to give a smooth landing area.

2004: The Quick Pass Problem

"Quick Pass" systems are increasingly appearing to reduce people's time waiting in line, whether it is at tollbooths, amusement parks, or elsewhere.

Consider the design of a Quick Pass system for an amusement park. The amusement park has experimented by offering Quick Passes for several popular rides as a test. The idea is that for certain popular rides you can go to a kiosk near that ride and insert your daily park entrance ticket, and out will come a slip that states that you can return to that ride at a specific time later. For example, you insert your daily park entrance ticket at 1:15 P.M., and the Quick Pass states that you can come back between 3:30 and 4:30 P.M. when you can use your slip to enter a second, and presumably much shorter, line that will get you to the ride faster. To prevent people from obtaining Quick Passes for several rides at once, the Quick Pass machines allow you to have only one active Quick Pass at a time.

You have been hired as one of several competing consultants to improve the operation of Quick Pass. Customers have been complaining about some anomalies in the test system. For example, customers observed that in one instance Quick Passes were being offered for a return time as long as 4 hours later. A short time later on the same ride, the Quick Passes were given for times only an hour or so later. In some instances, the lines for people with Quick Passes are nearly as long and slow as the regular lines.

The problem then is to propose and test schemes for issuing Quick Passes in order to increase people's enjoyment of the amusement park. Part of the problem is to determine what criteria to use in evaluating alternative schemes. Include in your report a nontechnical summary for amusement park executives who must choose between alternatives from competing consultants.

Comments

The Outstanding papers were by teams from Harvard University, Merton College of the University of Oxford, Harvey Mudd College, University of Colorado, and University of Washington. Their papers were published in *The UMAP Journal* 25 (3) (2004): 281–354. The Ben Fusaro Award went to a team from MIT.

Problem Origin

The problem was contributed by Jerrold R. Griggs (University of South Carolina).

The 2005 Problems

The Flood Planning Problem

Lake Murray in central South Carolina is formed by a large earthen dam, which was completed in 1930 for power production. Model the flooding downstream in the event there is a catastrophic earthquake that breaches the dam.

Two particular questions:

- Rawls Creek is a year-round stream that flows into the Saluda River a short distance downriver from the dam. How much flooding will occur in Rawls Creek from a dam failure, and how far back will it extend?
- Could the flood be so massive downstream that water would reach up to the S.C. State Capitol Building, which is on a hill overlooking the Congaree River?

Comments

The Outstanding papers were by teams from Harvey Mudd College, University of Saskatchewan (Canada), and University of Washington. The Ben Fusaro Award went to a team from McGill University (Canada).

The Tollbooth Problem

Heavily-traveled toll roads such as the Garden State Parkway, Interstate 95, and so forth, are multi-lane divided highways that are interrupted at intervals by toll plazas. Because collecting tolls is usually unpopular, it is desirable to minimize motorist annoyance by limiting the amount of traffic disruption caused by the toll plazas. Commonly, a much larger number of tollbooths is provided than the number of travel lanes entering the toll plaza. Upon entering the toll plaza, the flow of vehicles fans out to the larger number of tollbooths, and when leaving the toll plaza, the flow of vehicles is required to squeeze back down to a number of travel lanes equal to the number of travel lanes before the toll plaza. Consequently, when traffic is heavy, congestion increases upon departure from the toll plaza. When traffic is very heavy, congestion also builds at the entry to the toll plaza because of the time required for each vehicle to pay the toll.

Make a model to help you determine the optimal number of tollbooths to deploy in a barrier-toll plaza. Explicitly consider the scenario where there is exactly one tollbooth per incoming travel lane. Under what conditions is this more or less effective than the current practice? Note that the definition of "optimal" is up to you to determine.

Comments

The Outstanding papers were by teams from Duke University, Harvard University, Massachusetts Institute of Technology, Rensselaer Polytechnic Institute, University of California–Berkeley (two teams), and University of Colorado. The Ben Fusaro Award went to one of the Outstanding teams from University of California–Berkeley