Journal of Computational Science 92 (2025) 102728

Contents lists available at ScienceDirect

Journal of Computational Science

journal homepage: www.elsevier.com/locate/jocs

Efficient portfolio selection through preference aggregation with Quicksort and the Bradley–Terry model

Yurun Ge [a,b], Lucas Böttcher [c,d], Tom Chou [b,e], Maria R. D'Orsogna [a,b,*]

[a] Department of Mathematics, California State University at Northridge, Los Angeles, CA, USA
[b] Department of Computational Medicine, University of California, Los Angeles, Los Angeles, CA, USA
[c] Department of Computational Science and Philosophy, Frankfurt School of Finance and Management, Frankfurt a. M., Germany
[d] Laboratory for Systems Medicine, University of Florida, Gainesville, FL, USA
[e] Department of Mathematics, University of California, Los Angeles, Los Angeles, CA, USA

ARTICLE INFO

Keywords:
Collective intelligence
Social choice
Portfolio selection
Participatory budgeting
Agentic systems
Preference aggregation
Quicksort
Bradley–Terry model

ABSTRACT

Allocating limited resources to a set of alternatives with uncertain long-term benefits is a common challenge in innovation management, research funding, and participatory budgeting. Related problems arise in emerging applications such as ranking outputs of large language models and coordinating decisions in agentic systems. All settings include multiple agents tasked with estimating the true value of a potentially large number of alternatives. These estimates, or quantities derived from them, are then aggregated to select a final portfolio that maximizes overall benefit, ideally using efficient methods. Standard sorting algorithms are ill-suited as they do not account for uncertainties associated with each agent's estimate. Furthermore, the cognitive load on agents can be demanding, especially if the number of alternatives to evaluate is large. Building on the Quicksort algorithm and the Bradley–Terry model, we develop four new, efficient aggregation protocols based on agent-assigned win probabilities of pairwise comparisons that are then globally aggregated. The pairwise comparisons we introduce not only reduce cognitive load on agents, but lead to aggregation protocols that outperform existing ones, which we confirm via numerical simulations. Our methods can be combined with sampling strategies to further reduce the number of pairwise comparisons.

1. Introduction

The problem of allocating limited resources to projects that provide the greatest benefit to stakeholders arises in many decision-making contexts. When the long-term value of an alternative is difficult to assess, the evaluating agents will provide a broad distribution of estimates that must be efficiently aggregated. Common examples include members of an organization who are tasked with selecting new innovation projects with uncertain returns [1,2] or community stakeholders in participatory budgeting [3,4] who must decide which public projects deserve funding [5,6]. Similar problems arise in emerging applications, such as ranking outputs of large language models (LLMs) [7–9] and coordinating decisions in multi-agent or agentic systems [10]. In many settings, the number of projects under consideration is large and may result in a large cognitive load for evaluators. How can agents meaningfully compare and rank numerous alternatives when their information is incomplete or uncertain? Addressing this question requires methods that both reduce individual cognitive effort and enable efficient aggregation of preferences so that a high-value project portfolio can be selected. While our methods apply to a wide range of selection problems, we focus on project portfolio selection for concreteness.

The effectiveness of various aggregation methods such as voting, averaging, and expert delegation has been examined within social choice theory [11,12] and organizational decision-making [1,2,13]. The above methods assume that agents use their own direct estimates of project value. Ranked voting methods, like the Borda count [12,14], are based on each agent's ordered ranking of projects. While these perform well in portfolio selection with uniform project costs [2,15], the cognitive load on agents when ranking a large number of projects can be large.

In this paper, we develop four project evaluation and aggregation methods that involve pairwise comparisons of projects at the agent level. Specifically, the agents, who do not know the intrinsic value of the projects they are called to evaluate, compare pairs of projects. These comparisons are then used in conjunction with the well-known Quicksort algorithm [16] and the Bradley–Terry model [17,18] to

* Corresponding author at: Department of Mathematics, California State University at Northridge, Los Angeles, CA, USA.
E-mail address: dorsogna@csun.edu (M.R. D'Orsogna).

https://doi.org/10.1016/j.jocs.2025.102728
Received 30 January 2025; Received in revised form 8 September 2025; Accepted 3 October 2025
Available online 7 October 2025

collectively rank projects so that a high-value project portfolio can be assembled.

We show that our proposed aggregation methods, based on pairwise comparison rules, outperform those that utilize value estimates or ordered rankings. Our findings are important because pairwise comparisons not only yield better outcomes but help reduce the cognitive burden of directly ranking many projects and are more plausible. According to Miller's law [19], human short-term memory is limited to processing about seven items at a time, making direct ranking increasingly unrealistic as the number of projects grows. Furthermore, pairwise comparisons are particularly useful when direct estimations are difficult due to psychological biases [20].

Aggregation methods based on pairwise comparisons remain relatively underexplored, particularly in contexts that involve uncertainty [21]. Algorithms for sorting under noisy information were only recently introduced [22,23] with some extensions enabling parallel processing [24]. Existing strategies to handle "dirty" comparisons often combine noisy data with a limited number of accurate, "clean" comparisons [25,26], which can be adjusted based on noise levels [27]. In some cases, approximate rankings can be achieved with relatively few comparisons [28]. Other works have modeled decision-making as an analytic hierarchy process [29] that includes pairwise comparisons and where fuzzy logic is incorporated to represent uncertainty [30]. Pairwise-comparison algorithms have also been proposed in machine learning for efficient item ranking [31,32]. Our work contributes to this nascent literature by developing efficient and easy-to-implement algorithms.

Since we combine aspects of the Bradley–Terry model with portfolio selection theory, the next two sections contain a concise overview of each topic, highlighting the elements most relevant to our study. In particular, in Section 2, we review the Bradley–Terry model and the pairwise-comparison algorithm that we later use when agents are tasked with comparing project pairs. In Section 3, we discuss how agents evaluate projects based on their expertise and project type, and where pairwise comparisons are performed according to the Bradley–Terry model. Various methods for aggregating the heterogeneous project evaluations are discussed in Section 4, including two existing methods that do not use pairwise comparisons (the Arithmetic Mean and the Borda Count) and four novel methods based on pairwise comparisons that use the Quicksort algorithm and the Bradley–Terry model. Limitations and advantages of all methods used are discussed in Section 5. In Section 6, we introduce the performance measure used to compare each of the portfolios generated by the six aggregation methods. Performances are evaluated numerically in Section 7 for various parameter choices: several of our proposed aggregation methods are shown to outperform existing ones. Sampling techniques to reduce the number of comparisons are also presented. Finally, in Section 8, we summarize and discuss our findings.

2. The Bradley–Terry model

The Bradley–Terry model is a statistical method for ranking n items based on repeated pairwise comparisons originally introduced to rank players using tournament outcomes [17,18]. Due to its versatility, the Bradley–Terry model has been applied to sports rankings, electoral preferences, skill-based matchmaking, psychological research, and in other domains where relative comparisons are more practical than standalone evaluations. More recently, Bradley–Terry models have also been used in machine learning [33], to help evaluate LLM outputs [8,34], and in other problems involving human choice [35–38].

The Bradley–Terry model assumes that outcomes of pairwise comparisons between the items to be ranked are known and that each item has an underlying latent "strength". These latent strengths are estimated by maximizing the likelihood of the given pairwise comparisons, typically using iterative algorithms [39–41]. Extensions include allowing items to be tied [42,43], multiple, rather than pairwise, comparisons [44], incorporating ordering-based advantages, such as playing on one's home-field in sports [45,46], or using only subsets of comparisons [47].

In the original formulation of the Bradley–Terry model, the items are n competing players and the pairwise-comparison outcomes w_{ij} are the number of times player i wins over player j. The latter are also referred to as win numbers. In our setting, we adapt the model by replacing the players with n "competing" projects (since not all can be selected) and by considering, instead of win counts, the probability w^{ℓ}_{ij} that agent ℓ prefers project i over project j. For concreteness, the exposition that follows illustrates how the latent strengths are obtained in the original context of the Bradley–Terry model. We later adapt the procedure to our specific project-selection setting.

Mathematically, the latent strengths of items (players or projects) i and j are denoted by π_i and π_j, respectively; the relative frequency that i wins over j is $\pi_i/(\pi_i+\pi_j)$. Given the win numbers w_{ij}, and denoting the strength parameter vector $\boldsymbol{\pi} = (\pi_1, \dots, \pi_n)^{\top}$, one wishes to maximize the log-likelihood function

$$
\begin{aligned}
l(\boldsymbol{\pi}) &= \sum_{i\neq j} w_{ij} \ln\left(\frac{\pi_i}{\pi_i+\pi_j}\right) \\
&= \sum_{i\neq j} w_{ij}\left[\ln(\pi_i) - \ln(\pi_i+\pi_j)\right].
\end{aligned}
\tag{1}
$$

Maximizing Eq. (1) involves iterative updates of the vector $\boldsymbol{\pi}$. Under certain conditions, this maximization has a unique solution [17]. In practice, for all i, one can differentiate $l(\boldsymbol{\pi})$ with respect to π_i and set the resulting expression to zero, leading to the implicit form

$$
\pi_i = \frac{\sum_{j\neq i} w_{ij}}{\sum_{j\neq i}\left(\frac{w_{ij}+w_{ji}}{\pi_i+\pi_j}\right)}. \tag{2}
$$

Thus, the updated strength of item i, $\tilde{\pi}_i$, is determined via

$$
\tilde{\pi}_i = \frac{\sum_{j\neq i} w_{ij}}{\sum_{j\neq i}\left(\frac{w_{ij}+w_{ji}}{\pi_i+\pi_j}\right)}, \tag{3}
$$

where π_i and $\pi_{j\neq i}$ are the strength parameters prior to the update. Iterations are repeated until convergence is reached and $\tilde{\pi}_i \approx \pi_i$. While this scheme is simple, it can be slow to converge. A more recent approach is Newman's method [41]. It is based on the update

$$
\tilde{\pi}_i = \frac{\sum_{j\neq i}\left(\frac{w_{ij}\pi_j}{\pi_i+\pi_j}\right)}{\sum_{j\neq i}\left(\frac{w_{ji}}{\pi_i+\pi_j}\right)}, \tag{4}
$$

which converges faster than the one in Eq. (3) by a factor of 3 to 100. Convergence speed and stability can be further improved by including the updated values after each iteration as in the Gauss–Seidel method [48], leading to

$$
\tilde{\pi}_i = \frac{\sum_{j\neq i}\left(\frac{w_{ij}\tilde{\pi}_j}{\pi_i+\tilde{\pi}_j}\right) + \sum_{j>i}\left(\frac{w_{ij}\pi_j}{\pi_i+\pi_j}\right)}{\sum_{j\neq i}\left(\frac{w_{ji}}{\pi_i+\tilde{\pi}_j}\right) + \sum_{j>i}\left(\frac{w_{ji}}{\pi_i+\pi_j}\right)}. \tag{5}
$$

Once the strength π_i of each item i is determined from Eq. (5), the vector $\boldsymbol{\pi}$ is used to generate a global ranking. The strength parameters can become ill-defined if an item never wins or never loses in the pairwise comparisons. For example, consider items 1, 2, and 3 with the following pairing results: 1 wins against 2, 1 wins against 3, 2 wins against 3; in this case the algorithm leads to π_1 diverging to infinity at a faster rate than π_2, while π_3 converges to 0. These scenarios, however, become rarer as the number of items increases.

In our portfolio-selection context, each agent ℓ evaluates n projects and performs comparisons for all distinct project pairs i, j with $i, j \in$

$\{1, \ldots, n\}$ and $i \neq j$. Agents do not know the exact value of the projects they are comparing and can only estimate these values. This uncertainty will propagate to the win number, rendering it a probability. In the next section, we describe how the "win probability" w^{ℓ}_{ij} that project i is better than project j according to agent ℓ is specifically constructed. In Section 4, the win probabilities w^{ℓ}_{ij} are aggregated over all agents, and a collective win probability w'_{ij} is derived. The set of all w'_{ij} are then used in conjunction with the Bradley–Terry model to assign latent strengths to all projects. To do this, we will use the improved Newman's method by setting $w_{ij} = w'_{ij}$ in the iterative scheme in Eq. (5).

3. Agent evaluations and win probabilities

In this section, we discuss how agents evaluate the n available projects and perform pairwise comparisons. Building on past work, we assume that the long-term values of the projects exist and are fixed, but cannot be precisely determined, leading to noisy evaluations [1,2]. Mathematically, each project $i \in \{1, \ldots, n\}$ is characterized by two parameters: its type $t_i \in [t_{\min}, t_{\max}]$ and value $v_i \in \mathbb{R}^+$. The value v_i defines the true (but unknown) benefit of project i over a specific time horizon, if chosen. This "ground truth" value may evolve or fluctuate over time due to societal shifts, environmental conditions, or complex interactions with other projects $j \neq i$. We do not consider these external sources of uncertainty in v_i and restrict ourselves to each agent's uncertainty in the estimation of v_i at the time of evaluation. This leads to subjective evaluations $v_{i\ell}$ (also referred to as perceived values) of project i from each agent $\ell \in \{1, \ldots, N\}$. To construct $v_{i\ell}$, we first assume that each agent ℓ involved in the decision-making process has a level of expertise $e_\ell \in [e_{\min}, e_{\max}]$ given by

$$e_\ell = e_{\mathrm{M}} - \frac{N + 1 - 2\ell}{N - 1}\beta. \tag{6}$$

According to Eq. (6) the e_ℓ values are evenly spaced across the interval $[e_{\min}, e_{\max}] := [e_{\mathrm{M}} - \beta, e_{\mathrm{M}} + \beta]$. Here, e_{M} represents the mean expertise level and β denotes the knowledge breadth that determines the expertise spread. For mathematical convenience, we set $e_{\mathrm{M}} = (t_{\min} + t_{\max})/2$ so that the mean expertise coincides with the mean project type. The expertise level distribution in Eq. (6) aligns with typical Hotelling-type models, where preferences are represented as distances along a line [49,50]. The values t_i and e_ℓ do not have any specific meaning; they are simply labels used to differentiate between various types and expertise levels. However, the alignment between t_i and e_ℓ affects the accuracy of agent ℓ's evaluation of project i's value, $v_{i\ell}$. Specifically, we assume that the noise $\eta_{i\ell} = v_{i\ell} - v_i$ follows a normal distribution centered at the origin with standard deviation $\sigma_{i\ell} = |t_i - e_\ell|$. That is, $\eta_{i\ell} \sim \mathcal{N}(0, \sigma^2_{i\ell})$, meaning that the closer the agent's expertise is to the project type, the lower the uncertainty. Each project is evaluated by N agents and their individual preferences are aggregated into a "collective" estimate. Since we assume resources are limited, we further impose that only a fixed number $n^* \leq n$ of projects can be included in the final portfolio. The collective estimate of each project determines whether or not it is part of the final selection.

We now allow agent ℓ to perform pairwise comparisons between projects i and j, with estimates $v_{i\ell}$ and $v_{j\ell}$, and uncertainties $\eta_{i\ell}$ and $\eta_{j\ell}$, respectively. The agent assigns a personal "win probability"

$$\begin{aligned} w^{\ell}_{ij} &:= \Pr\left(v_i > v_j\right) = \Pr\left((v_{i\ell} - \eta_{i\ell}) > (v_{j\ell} - \eta_{j\ell})\right) \\ &= \Pr\left((\eta_{i\ell} - \eta_{j\ell}) < (v_{i\ell} - v_{j\ell})\right) \end{aligned} \tag{7}$$

that project i is better than project j based on their evaluations $v_{i\ell}, v_{j\ell}$. The win probabilities w^{ℓ}_{ij} are later aggregated into a collective probability w'_{ij} that will be used to determine the relative strength of projects via Eq. (5). In this formulation, w^{ℓ}_{ij} is no longer a count of the number of times i wins over j as in Section 2, but the likelihood agent ℓ places on i winning over j given his or her evaluations and uncertainties. Similarly, w'_{ij} is the likelihood that collectively project i is deemed superior to project j. Under the assumption that the noise in the perceived value is independently and normally distributed, the difference $\eta_{i\ell} - \eta_{j\ell}$ follows a normal distribution, with mean zero and standard deviation $\sqrt{\sigma^2_{i\ell} + \sigma^2_{j\ell}}$. We thus rewrite Eq. (7) as

$$w^{\ell}_{ij} = \Phi\left(\frac{v_{i\ell} - v_{j\ell}}{\sqrt{\sigma^2_{i\ell} + \sigma^2_{j\ell}}}\right), \tag{8}$$

where Φ is the cumulative distribution function of the standard normal distribution. Eq. (8) quantifies the probability that agent ℓ deems project i to be better than project j. When the evaluation uncertainty vanishes, $\sigma_{i\ell}, \sigma_{j\ell} \to 0$, w^{ℓ}_{ij} is 1 for $v_{i\ell} > v_{j\ell}$ and zero otherwise, representing an indicator function for project i winning. An immediate consequence of Eq. (8) is that $w^{\ell}_{ij} = 1 - w^{\ell}_{ji}$.

Table 1 summarizes model variables and parameters used throughout this work. In the following section, we introduce six aggregation methods that, starting from the heterogeneous evaluations provided by the N agents, determine the $n^* \leq n$ projects to be included in the final portfolio. Two of these aggregation methods are standard and are based on the direct value estimates $v_{i\ell}$; the other four are contributions from this study and employ the win probabilities w^{ℓ}_{ij} in Eq. (8). We will show that our proposed aggregation methods, which use win probabilities, typically outperform those based on value estimates.

Table 1
Main model parameters. Unless otherwise stated, all parameters are real-valued.

Symbol	Description
$N \in \mathbb{Z}^+$	Number of agents
$n \in \mathbb{Z}^+$	Number of projects (or items)
$n^* \in \mathbb{Z}^+,\ n^* \leq n$	Budget constraint
$i, j \in \{1, \ldots, n\}$	Project label
$\ell \in \{1, \ldots, N\}$	Agent label
$v_i \in \mathbb{R}^+$	Value of project i
$t_i \in [t_{\min}, t_{\max}]$	Type of project i
$e_\ell \in [e_{\min}, e_{\max}]$	Expertise of agent ℓ
$\beta \geq 0$	Knowledge breadth of agents
e_{M}	Mean expertise level; $e_{\mathrm{M}} = (t_{\min} + t_{\max})/2$
$v_{i\ell}$	Value of project i, evaluated by agent ℓ
$\eta_{i\ell} = v_{i\ell} - v_i$	Noise of value of project i, associated with agent ℓ
$\sigma_{i\ell} > 0$	Uncertainty in value of project i, associated with agent ℓ
v'_i	Aggregate value of project i over all N agents
$w^{\ell}_{ij} \in (0, 1)$	Win probability of project i over project j from agent ℓ
$W^{\ell} \in (0, 1)^{n \times n}$	Matrix of all win probabilities w^{ℓ}_{ij} from agent ℓ
$w'_{ij} \in (0, 1)$	Aggregated probability of project i winning over project j
$W' \in (0, 1)^{n \times n}$	Matrix of all aggregated win probabilities w'_{ij}

4. Aggregation methods and portfolio selection

Once the individual inputs (projects evaluations $v_{i\ell}$, or win probabilities w^{ℓ}_{ij}) are known, the challenge is to aggregate them into a collective output from which the $n^* \leq n$ most desirable projects can be selected. The optimal aggregation of inputs is a well-studied topic in voting, social choice, and organizational decision-making, with various methods having been proposed. These include equal weighting, delegation to experts, majority rule and subgroup biasing [1,2,13]. Additional considerations such as the presence of hierarchies [51], guaranteeing system legitimacy and fairness [52,53], avoiding polarization [54] or budget constraints [15], may also influence the choice of aggregation method. Fig. 1 presents a schematic of the complete portfolio selection model we use in this work.

We proceed by illustrating the six aggregation methods used in this work. Of these six, the first two are existing ones based on direct evaluations or scores, the other four are novel to this study and use pairwise comparisons based on the Quicksort algorithm on the Bradley–Terry model by incorporating the win probabilities in Eq. (8). This allows to bypass using the direct project evaluations $v_{i\ell}$. Of the four novel methods, two rely on pairwise comparisons between all project

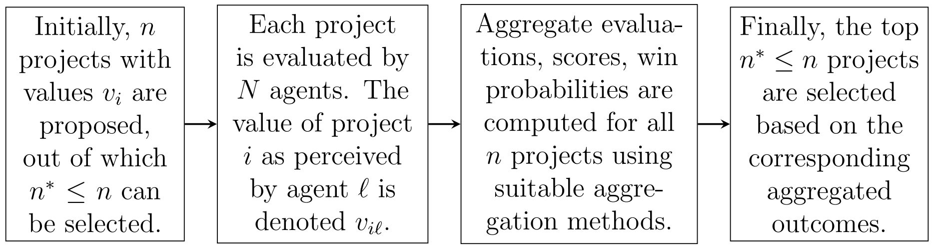

Fig. 1. Flowchart of the collective portfolio selection process. A set of n projects are proposed and evaluated by N agents. Aggregated evaluations, scores, or win probabilities are computed, and the top $n^* \leq n$ projects are selected based on these aggregated outcomes.

pairs, while the other two use comparisons restricted to a subset of projects.

4.1. Aggregation through direct evaluations or scores

(a) Arithmetic mean. This method uses the project estimates $v_{i\ell}$ from all N agents and averages them to obtain the aggregated value

$$v_i' = \frac{1}{N}\sum_{\ell=1}^{N} v_{i\ell}. \tag{9}$$

The n^* projects with largest aggregate values v_i' are then selected. The Arithmetic Mean is the most natural aggregation method, in which all inputs are equally weighted. However, the direct value estimates $v_{i\ell}$ may be difficult to ascertain in practice, and outliers can easily bias the mean. Ranking-based methods, for example using the Borda count, may be more robust to outliers [2].

(b) Borda count. The Borda Count, introduced in the late 18th century, is a rank-based aggregation method in which each agent ℓ ranks the n projects in descending order according to their estimated values $v_{i\ell}$ [14]. For each project i, we denote its position in agent ℓ's preference list by $\mathrm{pos}_\ell(i)$. The aggregated score s_i for project i is then calculated as the sum of the reversed ranks across all N agents. That is,

$$s_i = \sum_{\ell=1}^{N}\big(n - \mathrm{pos}_\ell(i)\big). \tag{10}$$

The n^* projects with the highest aggregated scores are selected for inclusion in the collective portfolio. This method is particularly robust against mis-classification and often outperforms the Arithmetic Mean, especially in conditions of high uncertainty [2].

4.2. Aggregation through pairwise comparisons

(c) Quicksort. Quicksort is a widely used sorting algorithm that uses a divide-and-conquer approach to sort items [16]. Its average-case time complexity is $\mathcal{O}(n\log(n))$, making it one of the most efficient sorting algorithms [55]. Our adaptation of Quicksort for project selection is presented in Algorithm 1. Quicksort selects a "pivot" project from the middle of the list of available projects and partitions the remaining ones into two sublists: one containing projects ranked worse than the pivot, and the other containing projects ranked better than or equal to the pivot. This partitioning process is recursively applied to each sublist. In our approach, we calculate the aggregated win probability w_{ij}' associated with projects i and j as

$$w_{ij}' = \frac{1}{N}\sum_{\ell=1}^{N} w_{ij}^{\ell}, \tag{11}$$

where w_{ij}^{ℓ} is given in Eq. (8), and consider project i to be better than the pivot p if the aggregated win probability of project i against the pivot p is at least 0.5, *i.e.* if $w_{ip}' \geq 0.5$. The Quicksort method produces a list of ranked projects based on their aggregated win probabilities, from which the best n^* are selected.

Algorithm 1 Quicksort with aggregated win-probability matrix

```
Require: Aggregated win-probability matrix W′ of size n × n
Ensure: Sorted index array idx
idx ← list of integers from 0 to n − 1
function Partition(low, high)
i ← low − 1
for j ← low to high − 1 do
    if W′[idx[j], idx[high]] < 0.5 then
        i ← i + 1
        Swap(idx[i], idx[j])
    end if
end for
Swap(idx[i + 1], idx[high])
return i + 1
end function
function QuickSortRecursive(low, high)
if low < high then
    pi ← Partition(low, high)
    QuickSortRecursive(low, pi − 1)
    QuickSortRecursive(pi + 1, high)
end if
end function
QuickSortRecursive(0, n − 1)
return idx
```

(d) Bradley–Terry method. Here, we build on the Bradley–Terry model described in Section 2 to aggregate the agent win probabilities w_{ij}^{ℓ} and to select the n^* projects to be included in the collective portfolio. The algorithm is as follows

- For each agent, the internal win probabilities w_{ij}^{ℓ} are used to construct a win-probability matrix $W^{\ell} \in (0,1)^{n\times n}$. Since there are N agents, there will also be N matrices W^{ℓ}.
- The aggregated win probabilities w_{ij}' are computed from Eq. (11) and W^{ℓ}. The aggregated values are used to construct the corresponding win-probability matrix $W' \in (0,1)^{n\times n}$. Win-probability aggregation methods in alternative to Eq. (11) may also be used.
- Newman's iteration is used to determine the relative strength of each project based on W' and Eq. (5).
- Projects are selected in descending order of relative strength until the desired number of projects n^* is reached. These are the ones that will be included in the collective portfolio.

4.3. Aggregation through sampled pairwise comparisons

Since it may not be feasible for agents to perform pairwise comparisons across all project pairs, the final two aggregation approaches we propose include modified versions of Quicksort and of the Bradley–Terry Method that utilize only a subset of comparisons. Limiting the number of win probabilities w_{ij}^{ℓ} used is a cost-effective strategy, as each additional comparison requires resources. However, as we show

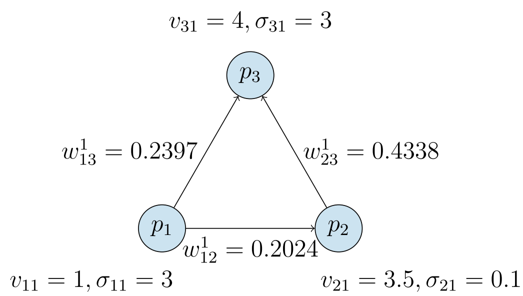

Fig. 2. Comparison of three projects p_1, p_2, and p_3 by a single agent. Each node represents a project and each directed edge represents a pairwise comparison between projects. Once w^1_{ij} has been determined, its complement w^1_{ji} is given by $w^1_{ji} = 1 - w^1_{ij}$.

in Section 7, choosing a good sampling protocol can also significantly enhance performance.

When using only a subset of pairwise comparisons, different samplings can influence the resulting rankings. For example, consider a single agent evaluating three projects p_1, p_2, p_3 with value estimates $v_{11} = 1$, $v_{21} = 3.5$, and $v_{31} = 4$ with uncertainties $\sigma_{11} = 3$, $\sigma_{21} = 0.1$, and $\sigma_{31} = 3$, respectively. Fig. 2 shows all pairwise comparisons by agent 1. Comparing project p_1 with project p_2 results in a win probability $w^1_{12} = 0.2024$, while comparing project p_2 with project p_3 yields $w^1_{23} = 0.4338$. This sequence leads to the ranking: $p_3 \succ p_2 \succ p_1$, where $x \succ y$ indicates that x is strictly preferred over y. However, if we compare project p_1 with projects p_2 and p_3, the win probabilities $w^1_{12} = 0.2024$ and $w^1_{13} = 0.2397$ produce a different ranking: $p_2 \succ p_3 \succ p_1$.

Although the ordering may depend on the specific subsample of comparisons, we explore an $\mathcal{O}(n)$ cyclic-graph sampling method, similar to what has been done in subgraph matching [56]. Our technique avoids performing all $\mathcal{O}(n^2)$ comparisons and can be visualized as extracting a subgraph from the complete graph generated by n projects. Given an ordered list of all p_i projects such as $(p_1, p_2, \dots, p_n)$, cyclic-graph sampling defines a structured subset of pairwise comparisons

$$((p_1, p_2), (p_2, p_3), \dots, (p_{n-1}, p_n), (p_n, p_1)), \tag{12}$$

where the parentheses contain pairs of projects to be compared. Outcomes depend on the initial $(p_1, p_2, \dots, p_n)$ ordering.

We now describe two additional aggregation methods that use cyclic graph sampling and a two-stage approach. In the first stage, an approximated project ranking is obtained through random sampling or by applying existing ranking algorithms. This preliminary ranking then serves as input to the second stage, where cyclic-graph sampling is used to refine the ranking via an optimization procedure based on the Bradley–Terry model. Specifically these methods are:

(e) Two-stage Bradley–Terry method.

- First stage: Generate an initial ranking where projects p_i ($i \in \{1, \dots, n\}$) are selected uniformly at random without replacement from the n available ones. Construct a cyclic-graph sampling of the randomly ordered list as shown in Eq. (12), and for each of the n pairs calculate w'_{ij} via Eq. (11). Then apply Newman's iteration given in Eq. (5) using the win probabilities w'_{ij} to obtain an approximate ranking.
- Second Stage: Starting from the approximate ranking, compute the corresponding win probabilities w'_{ij} using the cyclic-graph sampling in Eq. (12). To further refine the ranking, apply Newman's iteration again, using the win probabilities obtained in the first stage. Set win probabilities that are not calculated in either stage to 0.

(f) Two-stage Quicksort.

- First Stage: Instead of relying on randomly selected pairwise comparisons, apply the Quicksort algorithm to the matrix of aggregated win probabilities W' with elements w'_{ij} as shown in Eq. (11) to generate an initial ranking. Sample only the necessary entries of the aggregated win-probability matrix W' to keep a $\mathcal{O}(n\,\log(n))$ complexity. Since the underlying estimates are noisy observations, this Quicksort-derived ranking may deviate from the true ranking that would be obtained in the absence of uncertainty.
- Second Stage: Starting from the Quicksort ranking, compute the corresponding win probabilities w'_{ij} using the cyclic-graph sampling in Eq. (12). Then apply Newman's iteration given in Eq. (5) for a refined ranking. Unlike in the Two-Stage Bradley–Terry method discussed in (e), only consider win probabilities associated with the cyclic graph structure and not those obtained in the first stage.

5. Values, scores, or win probabilities?

We now discuss some of the advantages and limitations of the six aggregation methods and the quantities they rely on (*i.e.*, values, scores, and win probabilities). When outliers are present, aggregating win probabilities using Eq. (11) may be preferable to using the arithmetic mean in Eq. (9). To illustrate this, consider three agents evaluating Project 1 and Project 2. The first agent holds a highly favorable view of Project 1, while the other two agents assign lower value estimates to it. If the first agent's evaluation is an outlier – say if v_{11} approaches infinity – its influence on the aggregated outcome differs substantially between the two methods. Under the Arithmetic Mean, the aggregated value for Project 1, v'_1, is highly skewed by the outlier and may approach infinity as well. This disproportionate influence from a single agent distorts the collective assessment of Project 1's value. Win probabilities mitigate the impact of outliers, since they are bounded quantities. Let us assume that the extreme value from the first agent translates into a win probability of $w^1_{12} = 0.98$, indicating a strong preference. If the other two agents provide negative assessments of Project 1 with respect to Project 2, such as $w^2_{12} = w^3_{12} = 0.2$, the aggregated win probability, calculated using Eq. (11), results in $w'_{12} = 0.46$. This result is more closely aligned with the agents' evaluations compared to the outcome produced by the arithmetic mean.

Using win probabilities also offers an advantage over the Borda Count, as it more precisely captures individual preferences through real-valued probabilities. Consider two agents evaluating Project 1 and Project 2. The first agent strongly prefers Project 1 over Project 2 ($w^1_{12} = 0.8$), while the second agent only slightly favors Project 2 over Project 1 ($w^2_{12} = 0.46$). The aggregated win probability, $w'_{12} = 0.63$, indicates that Project 1 is the preferred choice overall, reflecting the stronger preference of the first agent. This approach takes into account the intensity of each agent's preference. On the other hand, if the Borda Count is used, each project would receive a Borda score of 1, resulting in a tie. This outcome fails to differentiate between the strong preference expressed by the first agent and the more moderate preference of the second.

6. Comparing aggregation methods

Once the six aggregation methods (a–f) have been used to assemble a final portfolio of $n^* \leq n$ projects, a natural question arises: which method is the most effective in selecting the most valuable projects? Recall that agents do not know the actual values v_i of the projects they are estimating and that their decision-making is based on project estimates $v_{i\ell}$ that can be quite different from the actual values or on scores or win probabilities that are also subject to uncertainty. The n^* projects that are selected for inclusion in the collective portfolio can thus include projects that were estimated to have high value (or that ranked high, or that had large win probabilities) but that in practice do not.

The effectiveness of a given aggregation method can be quantified in different ways. A performance metric for each method can be defined by either fixing or averaging over project types and/or fixing a set of agents or averaging over agent expertise distributions. Here, we choose to measure an aggregation method's performance by the expected total value $E(\beta; N, n, n^*)$ of $n^* \le n$ projects evaluated by N agents with knowledge breadth β. The expectation is taken over different project types, ensuring that the N agents (with knowledge breadth β) achieve robust performance on average across decision-making scenarios involving heterogeneous projects. The most effective aggregation method will be the one that yields the highest performance $E(\beta; N, n, n^*)$. In the absence of uncertainty, $E(\beta; N, n, n^*)$ simplifies to the total intrinsic value of the most valuable n^* projects for all aggregation methods. In the presence of uncertainty, $E(\beta; N, n, n^*)$ will depend on the chosen aggregation method.

As an example, consider $N = 3$ agents, each with a knowledge breadth $\beta = 0$, and $n = 3$ projects, from which $n^* = 2$ projects must be selected. The project values are $v_1 = 1$, $v_2 = 2$, and $v_3 = 3$. We assume that agents perceive the true project values (*i.e.*, $v_{i\ell} = v_i$ for $\ell \in \{1, 2, 3\}$). Since there is no uncertainty, agents will all select the two most valuable projects and the performance is calculated as $E(\beta = 0; N = 3, n = 3, n^* = 2) = v_2 + v_3 = 5$.

7. Simulation results

We now compare the performances $E(\beta; N, n, n^*)$ across the six aggregation methods (a–f) via numerical simulations and determine which of them yields the highest $E(\beta; N, n, n^*)$. Following prior work [1,2,15], we assume project types are uniformly distributed according to $\mathcal{U}(0, 10)$, so that $t_{\min} = 0$, $t_{\max} = 10$. The expertise value of the central decision maker is set at $e_M = (t_{\min} + t_{\max})/2 = 5$. We consider $n = 30$ projects, $N = 3$ agents, and a target of $n^* = 15$ projects. We define the value of project i as $v_i = i$ with $i \in \{1, \dots, 30\}$. The uncertainty in agent ℓ's project evaluations is quantified by additive Gaussian noise with zero mean and standard deviation $\sigma_{i\ell} = |t_i - e_\ell|$, where the expertise level e_ℓ of agent ℓ is given by Eq. (6). Variations in value distribution, type distribution, and other parameters are known to not significantly affect the relative ordering of aggregation-rule performance [2].

All our results are based on Monte Carlo simulations. For methods based on pairwise comparisons and win probabilities, we use 100,000 independent and identically distributed samples. For the remaining two methods, Arithmetic Mean and Borda Count, which are computationally less demanding, we increase the sample size to 500,000. The theoretical maximum performance is $\sum_{i=16}^{30} v_i = \sum_{i=16}^{30} i = 345$.

We consider two scenarios for computing the win probabilities w'_{ij}. In the first scenario, the probabilities are calculated according to Eqs. (8) and (11). However, in real-world applications of aggregation methods based on pairwise comparisons and win probabilities, assigning probabilities with several decimal places may be impractical. Therefore, in the second scenario, we prespecify a set of win probabilities from which agents can choose when making pairwise comparisons.

7.1. Continuous win probabilities

In Fig. 3, we plot the performance $E(\beta) \equiv E(\beta; N = 3, n = 30, n^* = 15)$ for the six aggregation methods (a–f) as a function of knowledge breadth β. Both the Arithmetic Mean and the Borda Count methods are known to be effective in identifying high-value projects within a portfolio [2]. In particular, the Borda Count is more robust to evaluation outliers than the Arithmetic Mean and performs better across a wide range of parameters. This is observed in Fig. 3, which also

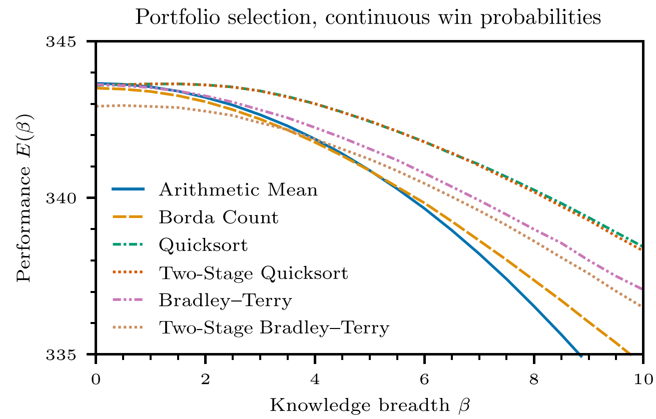

Fig. 3. Portfolio selection using continuous values of win probabilities. In this example, $n^* = 15$ projects are to be selected out of $n = 30$ projects by $N = 3$ agents. We show the performance $E(\beta) \equiv E(\beta; N, n, n^*)$ as a function of the knowledge breadth β for the six aggregation methods (a–f). The Quicksort (c) and Bradley–Terry (d) approaches proposed in this work perform favorably compared to the existing Arithmetic Mean (a) and Borda Count (b) methods, especially for larger knowledge breadth values $\beta \gtrapprox 5.5$, which are associated with larger uncertainties.

shows that Quicksort (c), Two-Stage Quicksort (f), and the Bradley–Terry method using all pairwise comparisons (d), outperform both the Arithmetic Mean (a) and Borda Count (b) methods, particularly for higher β. The Two-Stage Bradley–Terry method (e) performs worse than both the Arithmetic Mean (a) and Borda Count (b) for knowledge breadths $\beta \lessapprox 5.5$.

Since the above findings are based on aggregating the evaluations of only $N = 3$ agents we also conducted simulations for $N = 15$ and $N = 30$ agents to verify how absolute and relative performances would change upon increasing N. We found that the absolute performance of all six methods increases with N, while their relative performance remains similar. Thus, at least for modest N, Quicksort (c), Two-Stage Quicksort (f), and the Bradley–Terry method using all pairwise comparisons (d) emerge as high performing aggregation methods independent of the number of agents N. Additionally, the performance gap between the three methods above and the Two-Stage Bradley–Terry method widens with N. This is because compared to other methods, the Two-Stage Bradley–Terry method uses a sampling protocol that leaves more entries in the aggregated win probability matrix W' empty.

7.2. Discrete win probabilities

In practical applications of the Bradley–Terry method, it may be necessary to prespecify a set of win probabilities w^ℓ_{ij} from which agents can choose. Limiting probability values to a finite, manageable set, may be helpful in decision-making scenarios where high precision is not feasible. In Fig. 4, we show a comparison of the aggregation methods (a–f) where the win probabilities for methods (c–f) are restricted to values taken from a set of 11 possibilities given by $\{0.01, 0.1, 0.2, \dots, 0.8, 0.9, 0.99\}$. The relative performance ranking of the methods remains unchanged. However, the performance values of Quicksort and Two-Stage Quicksort exhibit a greater difference compared to the continuous case shown in Fig. 3. Recall that the Two-Stage Quicksort method employs a refinement stage in which the final ranking is computed according to Newman's iteration given in Eq. (5). While this second stage had little impact on performance in the continuous case, it substantially affects results when using the prespecified win probabilities listed above. This is consistent with the observation that Newman's iteration (or similar iterative methods used in the Bradley–Terry method) performs well even when rankings are derived from a limited set of tournament outcomes.

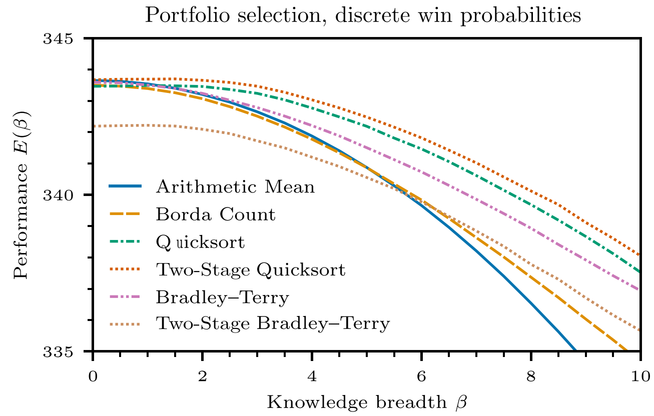

Fig. 4. Portfolio selection using discrete values of win probabilities. We show the performance $E(\beta) \equiv E(\beta; N, n, n^*)$ as a function of the knowledge breadth β for the six aggregation methods (a–f) using the same parameters as in Fig. 3. Here, agents that select projects using win-probability-based approaches, Quicksort (c), Two-Stage Quicksort (f), Bradley–Terry (d), and Two-Stage Bradley–Terry (e), are limited to selecting win probabilities from a discrete set of 11 values: $\{0.01, 0.1, 0.2, \dots, 0.8, 0.9, 0.99\}$. The Arithmetic Mean (a) and Borda Count (b) curves are the same as those in Fig. 3 and are included for reference. The relative ranking of the six methods remains unchanged compared to Fig. 3 in which continuous win probabilities are used. However, the difference in performance between Quicksort (c) and Two-Stage Quicksort (f) is more pronounced than in Fig. 3.

7.3. Efficiency and scalability

We now estimate the computational cost of each aggregation method. Aggregating inputs via the Arithmetic Mean does not involve any pairwise comparisons. However, a single sorting step is still necessary after aggregation to determine which of the $n^* \leq n$ projects should be included in the final portfolio; if this is done via the standard Quicksort algorithm, the number of necessary comparisons scales as $\mathcal{O}(n \log(n))$. If projects are instead aggregated via the Borda Count, each of the N agents must rank their own scores. This requires $\mathcal{O}(n \log(n))$ comparisons per agent to produce a ranking if the standard Quicksort algorithm is similarly used. For the other four methods that are based on win probabilities, the average number of pairwise comparisons are:

- Bradley–Terry (c) (all pairwise comparisons): $\mathcal{O}(n^2)$
- Quicksort (d) (without a second refinement stage): $\mathcal{O}(n \log(n))$
- Two-Stage Bradley–Terry (e): $\mathcal{O}(n)$
- Two-Stage Quicksort (f): $\mathcal{O}(n \log(n))$

In Fig. 5, we show the number of pairwise comparisons for the four methods above using discrete win probabilities. Since the number of pairwise comparisons depends on n and not on the set of possible win probability values, similar trends are observed for continuous win probabilities. The Bradley–Terry method (c) utilizes all $30(30-1)/2 = 435$ possible comparisons regardless of β. Similarly, the Two-Stage Bradley–Terry method (e) involves approximately 58 comparisons regardless of the value of β. For aggregation methods that utilize Quicksort instead a β dependence emerges: Quicksort (d) (without a second refinement stage), requires a number of pairwise comparisons that decreases from 265 for $\beta = 0$ to 193 for $\beta = 10$. The Two-Stage Quicksort (f) method results in slightly more comparisons, with 266 for $\beta = 0$ and 194 for $\beta = 10$.

The observed decrease in the number of pairwise comparisons results from the interplay between noise in the estimates and the divide-and-conquer nature of Quicksort. The algorithm compares elements against a pivot to split the dataset into two parts. The fastest completion of the algorithm is when the two parts contain an equal number of

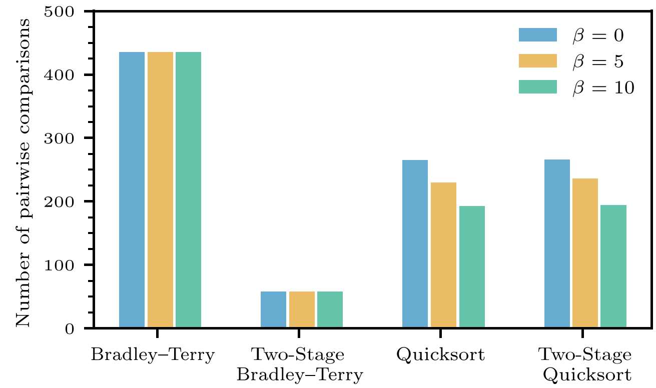

Fig. 5. Number of pairwise project comparisons across aggregation methods (c–f) for knowledge breadths $\beta \in \{0, 5, 10\}$ used to determine the performance $E(\beta)$ in Fig. 4 using discrete win probabilities. Estimates of the average number of pairwise comparisons for n are provided in Section 7.3. Although the Bradley–Terry (d), Quicksort (c), and Two-Stage Quicksort (f) methods achieve higher performance, the Two-Stage Bradley–Terry (e) method requires the least number of comparisons, making it the most practical choice for actual implementations.

projects, while the slowest is when one part contains all projects aside from the pivot, and the other contains none. As the noise in project values increases with β, it becomes less likely that highly imbalanced sublists arise during Quicksort recursions. While larger knowledge breadth allows Quicksort to be more efficient by requiring fewer number of pairwise comparisons, its overall performance decreases with β, but is still higher than or equal to other methods.

Although the Bradley–Terry, Quicksort, and Two-Stage Quicksort methods exhibit the highest performance in our simulations, the number of pairwise comparisons they require is likely too high for practical applications. In contrast, the Two-Stage Bradley–Terry method achieves favorable performance with significantly fewer comparisons, making it the most practical approach for real-world use. Further refinements could include identifying alternative sampling methods analogous to our cyclic graph approach that maintain strong performance while reducing the number of pairwise comparisons. Finding ways to sparsify the aggregate win probability matrix W' could also improve the applicability of our methods.

8. Conclusions

In this work, we compared six aggregation methods (a–f) for selecting project portfolios under uncertainty, including four novel ones based on pairwise comparisons (c–f). Agents evaluate projects without knowing their long-term value; the accuracy of these evaluations depends on how well agent expertise matches project types, with misalignment favoring large errors. How to arrive at a decision? Typically, direct estimations are collectively aggregated. However, when estimates are difficult to obtain or when expertise mismatches lead to outliers, Borda-type methods that rely on rankings offer a more robust alternative. Yet, generating full rankings can be cognitively demanding, especially when the number of projects is large.

To improve upon existing methods, we established a connection between portfolio selection, the Quicksort algorithm, and the Bradley–Terry model where project rankings are inferred from agent-specific win probabilities in pairwise comparisons. Based on this, we proposed four new aggregation methods. The first extends Quicksort to rank projects using aggregated win probabilities with a computational complexity of $\mathcal{O}(n \log(n))$; the second uses Newman's method with a complexity of $\mathcal{O}(n^2)$. To further lower the number of required comparisons, we incorporated a cyclic-graph sampling technique to both approaches, yielding two other aggregation methods of computational complexity $\mathcal{O}(n)$ (Two-Stage Bradley–Terry) and $\mathcal{O}(n\,\log(n))$ (Two-Stage Quicksort).

Our methods are relevant to participatory budgeting, social choice, organizational decision-making, and other resource allocation problems that involve decision-making under uncertainty. Our sampling and ranking methods can also be applied to rank foundation models such as LLMs.

Note that our analysis assumed uniform project costs, distributions over a single type variable, and a single set of project values. These simplifying assumptions can be relaxed to explore heterogeneous project costs and varying type and value distributions, providing a more realistic setting and additional insight into the determinants of performance in pairwise aggregation methods. Depending on context, it may also be useful to incorporate delegation strategies, querying only agents with relevant expertise, or to use alternative aggregation methods, such as the median instead of the mean. Exploring how agent interactions influence evaluations, for instance via social influence network models, could also offer valuable insights [57–59].

CRediT authorship contribution statement

Yurun Ge: Writing – original draft, Visualization, Methodology, Investigation, Formal analysis, Conceptualization. **Lucas Böttcher:** Writing – review & editing, Writing – original draft, Visualization, Supervision, Methodology, Investigation, Formal analysis, Conceptualization. **Tom Chou:** Writing – review & editing, Visualization, Investigation. **Maria R. D'Orsogna:** Writing – review & editing, Writing – original draft, Visualization, Supervision, Investigation, Funding acquisition, Formal analysis, Conceptualization.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Maria D'Orsogna reports financial support was provided by Army Research Office. Maria D'Orsogna reports equipment, drugs, or supplies was provided by National Science Foundation. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The authors thank Ronald Klingebiel for helpful discussions. This work was supported by the Army Research Office, United States through W911NF-23-1-0129 (YG, LB, MRD) and by the National Science Foundation, United States through grant MRI-2320846 (MRD). LB also acknowledges funding from hessian.AI.

Data availability

No data was used for the research described in the article.

References

[1] F.A. Csaszar, J. Eggers, Organizational decision making: An information aggregation view, Manag. Sci. 59 (10) (2013) 2257–2277.
[2] L. Böttcher, R. Klingebiel, Organizational selection of innovation, Organ. Sci. 36 (2025) 387–410.
[3] B. Wampler, A guide to participatory budgeting, in: A. Shah (Ed.), Participatory Budgeting, The World Bank, Washington, DC, 2007, pp. 21–54.
[4] H. Aziz, N. Shah, Participatory budgeting: Models and approaches, in: T. Rudas, G. Péli (Eds.), Pathways Between Social Science and Computational Social Science: Theories, Methods, and Interpretations, Springer International Publishing, Cham, Switzerland, 2021, pp. 215–236.
[5] G. Benade, S. Nath, A.D. Procaccia, N. Shah, Preference elicitation for participatory budgeting, Manag. Sci. 67 (2021) 2813–2827.
[6] D. Helbing, S. Mahajan, R.H. Fricker, A. Musso, C.I. Hausladen, C. Carissimo, D. Carpentras, E. Stockinger, J.A. Sanchez-Vaquerizo, J.C. Yang, et al., Democracy by design: Perspectives for digitally assisted, participatory upgrades of society, J. Comp. Sci. 71 (2023) 102061.
[7] G. Zhang, M. Hardt, Inherent trade-offs between diversity and stability in multi-task benchmark, 2024, arXiv preprint arXiv:2405.01719.
[8] W. Chiang, L. Zheng, Y. Sheng, A.N. Angelopoulos, T. Li, D. Li, B. Zhu, H. Zhang, M.I. Jordan, J.E. Gonzalez, I. Stoica, Chatbot arena: An open platform for evaluating LLMs by human preference, in: Forty-First International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024, 2024.
[9] A. Ghosh, S. Dziadzio, A. Prabhu, V. Udandarao, S. Albanie, M. Bethge, ONEBench to test them all: Sample-level benchmarking over open-ended capabilities, in: W. Che, J. Nabende, E. Shutova, M.T. Pilehvar (Eds.), Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Vienna, Austria, 2025, pp. 32445–32481.
[10] L. Wang, C. Ma, X. Feng, Z. Zhang, H. Yang, J. Zhang, Z. Chen, J. Tang, X. Chen, Y. Lin, et al., A survey on large language model based autonomous agents, Front. Comput. Sci. 18 (6) (2024) 186345.
[11] E. Elkind, P. Faliszewski, P. Skowron, A. Slinko, Properties of multiwinner voting rules, Soc. Choice Welf. 48 (2017) 599–632.
[12] F. Brandt, V. Conitzer, U. Endriss, J. Lang, A.D. Procaccia (Eds.), Handbook of Computational Social Choice, Cambridge University Press, Cambridge, UK, 2016.
[13] H.J. Einhorn, R.M. Hogarth, E. Klempner, Quality of group judgment, Psychol. Bull. 84 (1) (1977) 158.
[14] J.-C. de Borda, Mémoire sur les élections au scrutin, in: Histoire de L'Académie Royale des Sciences, 1781, pp. 657–665.
[15] Y. Ge, L. Böttcher, T. Chou, M.R. D'Orsogna, A knapsack for collective decision-making, 2024, arXiv:2409.13236.
[16] C.A. Hoare, Quicksort, Comput. J. 5 (1962) 10–16.
[17] E. Zermelo, Die berechnung der turnier-ergebnisse als ein maximumproblem der wahrscheinlichkeitsrechnung, Math. Z. 29 (1) (1929) 436–460.
[18] R.A. Bradley, M.E. Terry, Rank analysis of incomplete block designs: I. The method of paired comparisons, Biom. 39 (3/4) (1952) 324–345.
[19] G.A. Miller, The magical number seven, plus or minus two: Some limits on our capacity for processing information, Psychol. Rev. 63 (2) (1956) 81–97.
[20] W.S. Torgerson, Theory and Methods of Scaling, Wiley, New York, NY, 1958.
[21] U. Feige, P. Raghavan, D. Peleg, E. Upfal, Computing with noisy information, SIAM J. Comp. 23 (5) (1994) 1001–1018.
[22] M. Braverman, E. Mossel, Noisy sorting without resampling, in: Proceedings of the 19th Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2008, San Francisco, California, USA, January 20-22, 2008, SIAM, Philadelphia, PA, 2008, pp. 268–276.
[23] M. Braverman, E. Mossel, Sorting from noisy information, 2009, arXiv preprint arXiv:0910.1191.
[24] M. Braverman, J. Mao, S.M. Weinberg, Parallel algorithms for select and partition with noisy comparisons, in: Proceedings of the 48th Annual ACM Symposium on Theory of Computing, Association for Computing Machinery, New York, NY, 2016, pp. 851–862.
[25] X. Bai, C. Coester, Sorting with predictions, in: Proceedings of the 37th International Conference on Neural Information Processing Systems, NIPS '23, Curran Associates Inc., Red Hook, NY, 2024, pp. 26563–26584.
[26] P. Lu, X. Ren, E. Sun, Y. Zhang, Generalized sorting with predictions, in: 4th Symposium on Simplicity in Algorithms, SOSA, Society for Industrial and Applied Mathematics, Philadelphia, PA, 2021, pp. 111–117.
[27] T.H. Chan, E. Sun, B. Wang, Generalized sorting with predictions revisited, in: Frontiers of Algorithmics, in: Lecture Notes in Computer Science, vol. 13933, Springer International Publishing, Cham, Switzerland, 2023, pp. 29–41.
[28] J. Giesen, E. Schuberth, M. Stojaković, Approximate sorting, Fund. Inform. 90 (1–2) (2009) 67–72.
[29] E.H. Forman, S.I. Gass, The analytic hierarchy process: An exposition, Oper. Res. 49 (2001) 469–486.
[30] A. Emrouznejad, W. Ho, Fuzzy Analytic Hierarchy Process, Chapman and Hall, CRC Press, Boca Raton, FL, 2020.
[31] F. Wauthier, M. Jordan, N. Jojic, Efficient ranking from pairwise comparisons, in: Proceedings of the 30th International Conference on Machine Learning, in: Proceedings of Machine Learning Research, vol. 28, PMLR, Atlanta, Georgia, 2013, pp. 109–117.
[32] N. Ailon, An active learning algorithm for ranking from pairwise preferences with an almost optimal query complexity, J. Mach. Learn. Res. 13 (2012) 137–164.
[33] J. Wainer, A Bayesian Bradley-Terry model to compare multiple ML algorithms on multiple data sets, J. Mach. Learn. Res. 24 (341) (2023) 1–34.
[34] P.Y. We, J. Nagler, J.A. Tucker, S. Messing, Large language models can be used to estimate the latent positions of politicians, 2023, arXiv preprint arXiv:2303.12057v4.
[35] C. Muslimani, M.E. Taylor, Leveraging sub-optimal data for human-in-the-loop reinforcement learning, in: Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems, AAMAS '24, International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 2024, pp. 2399–2401.

[36] M. Peschl, A. Zgonnikov, F.A. Oliehoek, L.C. Siebert, MORAL: Aligning AI with human norms through multi-objective reinforced active learning, in: Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems, AAMAS '22, International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 2022, pp. 1038–1046.
[37] R. Herbrich, T.P. Minka, T. Graepel, TrueSkill™: A Bayesian Skill Rating System, in: Advances in Neural Information Processing Systems, Vol. 19, MIT Press, 2006.
[38] T.P. Minka, R. Cleven, Y. Zaykov, TrueSkill 2: An improved Bayesian skill rating system, 2018, URL https://api.semanticscholar.org/CorpusID:52906551.
[39] L.R. Ford, Solution of a ranking problem from binary comparisons, Am. Math. Mon. 64 (1957) 28–33.
[40] D.R. Hunter, MM algorithms for generalized Bradley-Terry models, Ann. Stat. 32 (2004) 384–406.
[41] M. Newman, Efficient computation of rankings from pairwise comparisons, J. Mach. Learn. Res. 24 (238) (2023) 1–25.
[42] P.V. Rao, L.L. Kupper, Ties in paired-comparison experiments: A generalization of the Bradley–Terry model, J. Amer. Statist. Assoc. 62 (1967) 194–204.
[43] R.R. Davidson, On extending the Bradley-Terry model to accommodate ties in paired comparison experiments, J. Amer. Statist. Assoc. 65 (1970) 317–328.
[44] R.N. Pendergrass, R.A. Bradley, Ranking in triple comparisons, in: I. Olkin, S.G. Ghurye, W. Hoeffding, W.G. Madow, H.B. Mann (Eds.), Contributions To Probability and Statistics: Essays in Honor of Harold Hotelling, Stanford University Press, Palo Alto, CA, 1960, pp. 331–351.
[45] A. Agresti, Categorical Data Analysis, Wiley, New York, NY, 1990.
[46] J.I. Marden, Analyzing and Modeling Rank Data, Chapman and Hall, London, UK, 1995.
[47] A. Liusie, V. Raina, Y. Fathullah, M. Gales, Efficient LLM comparative assessment: A product of experts framework for pairwise comparisons, in: Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Miami, FL, 2024, pp. 6835–6855.
[48] R. Kress, Numerical Analysis, Vol. 181, Springer Science & Business Media, Berlin, Germany, 2012.
[49] H. Hotelling, Stability in competition, Econ. J. 39 (153) (1929) 41–57.
[50] W. Novshek, Equilibrium in simple spatial (or differentiated product) models, in: A. Mas-Colell (Ed.), Noncooperative Approaches To the Theory of Perfect Competition, Academic Press, Cambridge, MA, 1982, pp. 199–212.
[51] S.E. Humphrey, J.R. Hollenbeck, C.J. Meyer, D.R. Ilgen, Hierarchical team decision-making, Res. Pers. Hum. Resour. Manag. 21 (2002) 175–213.
[52] C.I. Hausladen, R. Hänggli Fricker, D. Helbing, R. Kunz, J. Wang, E. Pournaras, How voting rules impact legitimacy, Humanit. Soc. Sci. Commun. 11 (1) (2024) 1–10.
[53] J.C. Yang, C.I. Hausladen, D. Peters, E. Pournaras, R. Hnggli Fricker, D. Helbing, Designing digital voting systems for citizens: Achieving fairness and legitimacy in participatory budgeting, Digit. Gov.: Res. Pr. 5 (3) (2024) 1–30.
[54] C. Alós-Ferrer, J. Buckenmaier, Voting for Compromises: Alternative Voting Methods in Polarized Societies, Vol. 394, Working Paper, University of Zurich, Department of Economics, 2021.
[55] T.H. Cormen, C.E. Leiserson, R.L. Rivest, C. Stein, Introduction to Algorithms, MIT Press, Cambridge, MA, 2022.
[56] Y. Ge, D. Yang, A.L. Bertozzi, Iterative active learning strategies for subgraph matching, Pattern Recognit. 158 (2025) 110797.
[57] A. Stomakhin, M.B. Short, A.L. Bertozzi, Reconstruction of missing data in social networks based on temporal patterns of interactions, Inverse Problems 27 (11) (2011) 115013.
[58] L. Böttcher, J. Nagler, H.J. Herrmann, Critical behaviors in contagion dynamics, Phys. Rev. Lett. 118 (8) (2017) 088301.
[59] J.R. Zipkin, F.P. Schoenberg, K. Coronges, A.L. Bertozzi, Point-process models of social network interactions: Parameter estimation and missing data recovery, Eur. J. Appl. Math. 27 (3) (2016) 502–529.