SAMPLING FOR INFORMATIVE QUERY SELECTION IN CONSTRUCTIVE PREFERENCE ELICITATION

Anonymous authors
Paper under double-blind review

ABSTRACT

In Multi-Objective Combinatorial Optimization, the Constructive Preference Elicitation (CPE) framework has been proposed to learn objective importance from the Decision Maker's perspective through pairwise comparisons between candidate solutions. State-of-the-art methods draw these query pairs from a precomputed solution pool based on an explicit exploration and exploitation tradeoff, which depends on a tuned parameter. In the context of dueling bandits, Thompson-sampling approaches instead balance exploration and exploitation implicitly by sampling from a posterior distribution over the objective weights. We explore this concept in the context of CPE for the first time, with a new query selection criterion that samples objective weights from the current estimated belief to generate pairs of candidates. However, posterior sampling ensures only that each candidate is plausible under the current belief, without guaranteeing that their *comparison* will provide useful information. We therefore introduce DRES (Disagreement Reranked Ensemble Sampling), which combines this sampling mechanism with disagreement-based reranking, selecting the candidate query pair expected to provide the strongest learning signal. We evaluate DRES across three multi-objective combinatorial domains and showcase that it reduces both end-regret and the number of queries compared to the state-of-the-art.

1 INTRODUCTION

Combinatorial optimization problems are common in many industries, ranging from production scheduling to vehicle routing. Given their complex environments, it is often the case that they involve multiple objectives, ranging from production time to energy consumption (Gianessi et al., 2021; Foschini et al., 2026a). Since these objectives are usually conflicting, improving one worsens another, leading to multiple optimal solutions representing different trade-offs. A straightforward way to address such multi-objective problems is through *a priori* methods (Miettinen, 1998a). In these approaches, the sub-objectives are aggregated into a single linear objective function, typically using a weighted sum (Aghaei et al., 2015). The resulting model is then solved by a solver to obtain a solution that reflects the trade-offs encoded by the weighted sum.

Such functions require specifying a set of weights that capture the importance of each objective, as different sets of weights may lead to different solutions representing different trade-offs. However, requiring a Decision Maker (DM) to provide precise numerical values for these weights may be unrealistic (Mesquita-Cunha et al., 2023). This motivates *(inter)active* methods, which instead aim to infer the objective weights by interacting with the DM. Among the most widely used are methods based on pairwise comparisons, which are known for keeping the cognitive load on the DM relatively low (Defresne et al., 2025).

In Active Learning, well-established pairwise preference elicitation approaches typically assume that candidate alternatives are available beforehand. This assumption does not hold in combinatorial settings, where explicitly computing the full set of optimal solutions is often impractical, as this set can be exponentially large (Bökler & Mutzel, 2015). As such, alternative approaches have been proposed for combinatorial settings such as the Constructive Preference Elicitation (CPE) framework (Teso et al., 2016). Recent work by Defresne et al. (2025) extends CPE with an easy-to-compute pool of candidate solutions, linking it more closely to pool-based Active Learning and enabling the investigation of methods from this field.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Among these methods are sampling-based approaches, which work by selecting candidate solutions based on the current belief about the DM's preferences, defined by some uncertainty model. In the combinatorial setting, these parameters directly map to the objective weights. Thompson sampling, in particular, is well suited to this setting because it has already been extended to pairwise preference elicitation through dueling posterior sampling (DPS) (Novoseller et al., 2019) and Self-Sparring (Sui et al., 2017). Furthermore, while the method from Defresne et al. (2025) requires tuning a trade-off parameter between exploration and exploitation, posterior sampling handles this balance implicitly.

Instead of searching for, or constructing, each query pair, DPS (Novoseller et al., 2019) and Self-Sparring (Sui et al., 2017) work by sampling independently twice over some belief over candidates. These beliefs cover many uncertainty models coming from different frameworks, which differ in their *representational capacity*. Introducing sampling-based query selection in CPE allows us to measure the impact of a more accurate representation of the DM's preference parameters on the active learning process, whether to focus efforts on investigating more representative uncertainty models or improving query selection criteria for larger combinatorial domains. Our empirical findings suggest that the latter warrants closer attention.

Sampling likely candidates does not ensure that the query itself is informative for the elicitation task. Novoseller et al. (2019)'s objective is to minimize the Bayesian cumulative regret over *both* candidates without guarantees about how informative the query, the pair, is. Information theoretic (Lindley, 1956; Houlsby et al., 2011a), decision theoretic (Viappiani & Boutilier, 2010; Vendrov et al., 2020), and query-by-committee (Seung et al., 1992; Defresne et al., 2025) all provide different measures to form informative query pairs. Christiano et al. (2023) and Lee et al. (2021) propose to rerank several query pairs but form these pairs through uniformly sampling the candidate pool, which is informative but not representative. However, none of these have been studied in the context of CPE.

The main contributions of this paper are as follows: 1) We propose DRES (Disagreement Reranked Ensemble Sampling), a new sampling-based query selection strategy for CPE. It combines sampling over the DM's preference parameters with a disagreement-based rerank step. 2) We evaluate how different uncertainty models behave in CPE by fixing the query selection criterion. We measure the representational capacity and evaluate its effects for preference elicitation in combinatorial domains. 3) We evaluate DRES empirically across three CPE problems. The results show that DRES outperforms the current state-of-the-art on both end-regret and number of queries.

2 Related work

Constructive Preference Elicitation computes optimal solutions in large combinatorial spaces based on user feedback (Dragone et al., 2018). Early approaches, like the minimax-regret criterion (Braziunas & Boutilier, 2012) or the Choice Perceptron (Dragone et al., 2018) assume implicit solution pools, relying on combinatorial optimization solvers to generate feasible solutions, since enumerating the Pareto front is often computationally expensive. Other methods that rely on explicit solution pools, like the Value of Information criterion (Guo & Sanner, 2010) and the *cluster* acquisition strategy (Defresne et al., 2025). The latter scores solutions based on an Upper Confidence Bound (UCB) score over the solution pool, removing the need for combinatorial solver calls during query selection. To reduce indifference between two proposed solutions, The authors first cluster the solution pool and pick the top UCB solution from two uniformly sampled clusters. But UCB scoring requires explicitly modeling exploration.

Dueling bandits learn preferences from noisy relative feedback, such as pairwise comparisons, instead of absolute rewards (Yue et al., 2012). Query pairs are formed by UCB (Zoghi et al., 2014), Interleaved Filter selection (Yue et al., 2012), Beat-the-Mean (Yue & Joachims, 2011), double Thompson sampling (Wu & Liu, 2016), and others. Later work extends this feedback from single arms to comparisons between teams of arms (Cohen et al., 2021) and to rankings over played subsets (Saha & Gopalan, 2019). Huang et al. (2024)'s method combines dueling bandits and evolutionary search for query selection and pool generation in continuous multi-objective optimization.

Implicit exploitation schedules rely on the uncertainty models' belief to provide a tradeoff between exploration and exploitation of their knowledge. Thompson (1933) sampling is often used in Bayesian inference (Russo et al., 2020), information theory (Russo & Roy, 2016), and bandits

literature (Wu & Liu, 2016; Wang & Chen, 2022; Agrawal & Goyal, 2014). Some uncertainty models are intractable, so Ensemble Sampling (ES) (Lu & Van Roy, 2017; Qin et al., 2023) is used to approximate Thompson sampling.

Posteriors over preference parameters can be modelled with uncertainty models like multivariate Gaussian distributions (Agrawal & Goyal, 2014; Wang & Chen, 2022) or their Laplace approximation (Chu & Ghahramani, 2005), pSGLD (Li et al., 2015; Chiang et al., 2025), and preferential Normalizing Flow posteriors (Mikkola et al., 2024; Dinh et al., 2017) are often applied. The current state-of-the-art (SOTA) employs a non-Bayesian approximation of a Bradley & Terry (1952) posterior using an ensemble of linear models (Defresne et al., 2025). Selecting an effective posterior approximation method for sequential decision-making problems remains difficult: Riquelme et al. (2018)'s work benchmarks many different approximation methods and shows that their relative performance remains problem-dependent.

3 BACKGROUND

MOCO. A set of l variables of a CO problem is written as $\{x_1, \ldots, x_l\}$, where each x_i takes values from its own domain D_i. The set of all possible assignments is given by the Cartesian product $\mathcal{Y} = \prod_{i=1}^{l} D_i$. This unconstrained set typically contains assignments that are undesirable. To rule out such combinations, constraints are introduced to remove these infeasible combinations, yielding the set of feasible solutions $\mathcal{F} \subseteq \mathcal{Y}$. We associate one *sub-objective* function $\phi_i : \mathcal{Y} \to \mathbb{R}$ with each of the n objectives and write $\phi : \mathcal{Y} \to \mathbb{R}^n$ for the resulting *sub-objective vector* $\phi(y) = [\phi_1(y), \ldots, \phi_n(y)]$ (Keeney & Raiffa, 1993b), often referred to as the *image* of that solution.

Linear scalarization, utility and regret. Standard combinatorial solvers require a single objective to optimize. A common strategy is to aggregate the n sub-objectives of $\phi(\cdot)$ into a single scalar function, in which each ϕ_i is weighted by a coefficient w_i. The resulting *utility function*,

$$u(y, w) = \sum_{i=1}^{n} w_i \, \phi_i(y), \tag{1}$$

is defined for every $y \in \mathcal{Y}$. In a maximization context, positive-signed sub-objectives are maximized, and negated sub-objectives are minimized. A minimization context has this relationship reversed. This transformation makes the multi-objective structure implicit and reduces the problem to single-objective optimization, which a standard solver can handle. We call $w \in \mathbb{R}^n_{\geq 0}$ the weight vector (Keeney & Raiffa, 1993a). If all components of w are equal, every sub-objective contributes equally; setting $w_i = 0$ removes the i-th sub-objective from consideration.

To measure how much a decision maker, represented by their weight vector w^*, likes a given solution, we use the utility function $u(y, w^*) = \langle w^*, \phi(y) \rangle = \sum_{i=1}^{n} w_i^* \phi_i(y)$ (Keeney & Raiffa, 1993b). The optimal utility for any user w is calculated using (Weber, 2023)): $u^*(w) = \max_{y \in \mathcal{F}} u(y, w)$. When given a user vector w^*, we can find the solution that maximizes utility as $y^* = \arg\max_{y \in \mathcal{F}} u(y, w^*)$. This solution represents the perfect compromise for that user.

Bradley-Terry. Each comparison reveals information about the DMs underlying preferences: if the DM prefers y_1 over y_2, written $y_1 \succ y_2$ (equivalently $y_2 \prec y_1$), we obtain partial information about the user's true utility function. We call y_+ the preferred solution and y_- the other half of the pair. Following utility theory (Keeney & Raiffa, 1993b), we seek w such that $u(y_+, w) > u(y_-, w)$. This relationship is often modeled probabilistically using the Bradley & Terry (1952) model. The likelihood of the pairwise preference of y_+ over y_- given a user weight w:

$$p(y_+ \succ y_- | w) = \frac{e^{\lambda u(y_+, w)}}{e^{\lambda u(y_+, w)} + e^{\lambda u(y_-, w)}} = \sigma\Big(\lambda\big(u(y_+, w) - u(y_-, w)\big)\Big), \tag{2}$$

where $\sigma(\cdot)$ is the sigmoid function and λ, the inverse temperature that governs the oracle's deterministic behavior.

Ensemble of linear models. Defresne et al. (2025)'s ensemble of linear models $\mathbf{W} \in \mathbb{R}^{M \times D}$, where each $w_i \in \mathbb{R}^D$ represents an independently initialized point estimate in weight space. Given

Algorithm 1 DRES query selection

Require: particle set $\boldsymbol{\mathcal{P}}^t \in \mathbb{R}^{M \times D}$, pool image $\boldsymbol{\Phi} \in \mathbb{R}^{N \times D}$, candidate count K

1: $\mathbf{U} \leftarrow \boldsymbol{\Phi}(\boldsymbol{\mathcal{P}}^t)^\top$ ▷ $N \times M$ utilities
2: $f_m \leftarrow \arg\max_n U_{n,m}$ for each member m ▷ favorite solutions
3: $\mathcal{B} \leftarrow$ up to K distinct pairs $\{f_a, f_b\}$, $f_a \neq f_b$, with $a, b \sim \mathcal{U}\{1..M\}$
4: **for all** $\{y_1, y_2\} \in \mathcal{B}$ **do**
5: $\quad p_m \leftarrow \sigma\big((U_{y_1,m} - U_{y_2,m})/\tau\big)$ for each m
6: $\quad s(y_1, y_2) \leftarrow \mathrm{Var}_m[p_m]$
7: **end for**
8: **return** $\arg\max_{\{y_1,y_2\} \in \mathcal{B}} s(y_1, y_2)$

$\mathcal{D} = \{(y_+^t, y_-^t)\}_{i=1}^t$, the collected DM responses, they update each weight vector using the following maximum likelihood estimate (MLE) update:

$$w_{MLE}^{t+1} = w_{MLE}^t + \eta\sigma\big(u(y_-^t, w_{MLE}^t) - u(y_+^t, w_{MLE}^t)\big)\Delta^t \tag{3}$$

with η some fixed step size and $\Delta^t = \phi(y_+^t) - \phi(y_-^t)$, which encodes the difference in sub-objective values.

4 DRES: Disagreement Reranked Ensemble Sampling

We introduce a new method, DRES, which operates on any finite set of weight vectors $\boldsymbol{\mathcal{P}}^t = [\mathbf{w}_1, \ldots, \mathbf{w}_M]^\top \in \mathbb{R}^{M \times D}$, which we call *particles*. Defresne et al. (2025)'s ensemble of linear models $\mathbf{W}$ is the most directly related, as it is a collection of independent point estimates. Likewise, the particles from MCMC and variational inference methods are interpretable as a comparable collection (Liu & Zhu, 2018; Chau & Rasonyi, 2020; Zhang & Li, 2025). Continuous probabilistic uncertainty models are less directly applicable.

It is well known that a finite collection of samples generated by any probability density function (PDF) may serve as partial information about that original PDF. For example, having more posterior draws reduces the variance of the model parameters found by methods like maximum-likelihood estimation (MLE). We therefore assert that a set of independent samples from continuous probabilistic distributions may serve as a set of particles, representing plausible weights of that distribution.

4.1 Implementation

Algorithm 1 requires a set of particles $\boldsymbol{\mathcal{P}}^t$ at time t, the images of the precomputed solution pool $\boldsymbol{\Phi}$ and the number of query pairs to rerank K. We explain each line below, with details discussed in appendix section A.4.

The procedure starts off by computing the index of the utility-maximizing solution for each particle, called the favorite candidate vector f, which is calculated using the $\arg\max$ over the utilities $\mathbf{U}$ on line 2. Line 3 starts the pairwise ensemble sampling procedure. First we sample two integers, a and b, between 1 and M uniformly. These two numbers represent indices of rows in the weight matrix $\boldsymbol{\mathcal{P}}^t$. We then find the favorite solutions that correspond with those particle indices. We only add the pair of solutions to $\mathcal{B}$ after we check if the pair was not already in $\mathcal{B}$ or if the corresponding solutions f_a and f_b are identical. We do this until the set of pairs has a size K or until we have exhausted some attempt count.

After collecting potential solution pairs, we calculate the disagreement over the Bradley & Terry (1952) choice probabilities by calculating the variance over p_m, the probability p_m each particle m assigns to preferring y_1 over y_2 using equation 2. Line 8 presents the pair that maximizes the variance to the DM.

4.2 ALR-parametrization

A weight vector represents some compromise over different objectives. We select the solution from the solution pool based on the relative differences in weightings, not the scale of that weighting. This

weight-space can therefore be interpreted as scale-invariant or compositional. The belief we require is compositional, whereas many approximate Bayesian techniques were defined in $\mathbb{R}^D$, which negatively impacts their ability to explore the weight-space. We address this at the model parametrization level, ensuring that each approximate-Bayesian user model is fitted using the same unaltered algorithm as its original publication. We do so by using the additive log-ratio (ALR) transform, explained in Appendix Section A.2, which is the standard transformation for compositional data Aitchison (1986). By convention, the final component w_D serves as a reference. The bijection between the open simplex and $\mathbb{R}^{D-1}$ is given by:

$$u_j = \ln\left(\frac{w_j}{w_D}\right), \quad j = 1, \ldots, D-1, \qquad\qquad w(u) = \operatorname{softmax}\big([u_1, \ldots, u_{D-1}, 0]\big). \tag{4}$$

One important note is that a weight vector with $w_i = 0$ has no ALR image, since $\ln 0$ is undefined. The faces of the simplex are approached only in the limit $\|u\| \to \infty$.

5 Experiments

We empirically evaluate the following research questions: 1) How does the representational capacity of a user model relate to the end-regret for ensemble sampling? 2) What is the impact of reranking the sampled query pairs? 3) What is the impact of ensemble size on elicitation quality?

5.1 CPE tasks

PC-TSP. We first consider a multi-objective prize-collecting traveling salesman problem (PC-TSP) (Véjar et al., 2024). This optimization problem tries to find a sub-tour of cities where one can visit each city at most once. The graph $G = (V, E)$ contains edges and vertices, representing roads and cities. A city is identified by a unique natural number $1, \ldots, |V|$, with the i-th having a penalty γ^i if it was not visited, which is indicated using o_i. Each edge $(i, j) \in E \subset \mathbb{N} \times \mathbb{N}$ has p properties, v_p^{ij}, over which the DM elicits a preference. We fix sub-tour length to 10 and 20.

The multi-objective optimization:

$$\min_y \sum_{(i,j)\in E} \big(w_1 v_1^{ij} + \cdots + w_p v_p^{ij}\big) y_{ij} + w_{p+1} \sum_{i \in V} \gamma^i (1 - o_i). \tag{5}$$

Relaxed solutions are permutations of all these identifiers (Defresne et al., 2025).

Computer configuration. Our second problem elicits preferences over configuration of personal computers (Guo & Sanner, 2010), where the MOCO problem is modeled after a dataset of 120 real configurations (McGinty & Smyth, 2003). Let $y_{c,v} \in \{0, 1\}$ be 1 if the component type $c \in \{1, \ldots 6\}$ the component instance $v \in D_c$ is taken, 0 otherwise. When we enumerate the one-hot vector and look at the k'th bit, its value will be written as $y_{c(k),v(k)}$. Each property has an associated cost, denoted cost_k. The final sub-objective value is the DM's acceptance or rejection of the total cost, and it is weighted using w_{77}. The cost will be normalized by dividing the cost value by the maximum price of any item in that type, denoted as $\max_c$. The range of the costs is thus reduced to the range of $[0, 1]$. Using linear scalarization, the MOCO problem is maximized for:

$$\max_y \sum_{k=1}^{76} w_k \cdot y_{c(k),v(k)} + w_{77} \cdot \sum_{k=1}^{76} \frac{\mathrm{cost}_k}{\max_{c(k)}} \cdot y_{c(k),v(k)}. \tag{6}$$

But only one value per property may be true, which is imposed by the following constraint:

$$\sum_{v \in D_c} y_{c,v} = 1 \quad \forall, c \in 1, \ldots, 6. \tag{7}$$

Any binary assignment of the variables can be interpreted as relaxed solutions to the MOCO problem (Defresne et al., 2025).

Land conservation. Our third problem is a combinatorial problem about conserving plots of land to save certain species, based on Billionnet (2021). All the problem instances and Pareto fronts come from Foschini et al. (2026b). Each instance is a 10×10 grid G of plots. Each plot $l \in G$ has a cost c_l and hosts a population of $p_{s,l}$, with $s \in S$ one of the 15 possible species. Let $y_l \in \{0, 1\}$ indicate that plot l is bought, and let $\mathrm{pop}_s(y) = \sum_{l \in G} p_{s,l}\, y_l$ be the protected population of species s; species s is saved when $\mathrm{pop}_s(y) \geq t_s$. The four sub-objectives are the number of species saved (ϕ_1), the budget spent (ϕ_2), the largest Manhattan distance d between two bought plots (ϕ_3), which keeps the bought plots close together, and the total protected population (ϕ_4), and are defined as:

$$\begin{aligned} \phi_1(y) &= \textstyle\sum_{s \in S} \mathbb{I}\big[\mathrm{pop}_s(y) \geq t_s\big], \quad \phi_2(y) = -\sum_{l \in G} c_l\, y_l, \\ \phi_3(y) &= -\max_{l,l' \in G} d(l, l')\, y_l\, y_{l'}, \qquad \phi_4(y) = \textstyle\sum_{s \in S} \mathrm{pop}_s(y), \end{aligned} \tag{8}$$

Using linear scalarization, the MOCO problems is:

$$\max_y \sum_{i=1}^{4} w_i\, \phi_i(y) \quad \text{s.t.} \quad \sum_{l \in G} c_l\, y_l \leq b, \qquad \sum_{l \in G} y_l \geq 1, \tag{9}$$

with $b = \lfloor \frac{1}{2} \sum_{l \in G} c_l \rfloor$. We used the precomputed Pareto front as the solution pool.

Nadir normalization over the solution pools. To ensure that the scales of different objectives do not influence the elicitation performance, we use Nadir-based normalization (Miettinen, 1998b). During the generation of the solution pool, we normalize the images of the solutions by:

$$\phi_i^{norm} = \frac{\phi_i}{y_i^{nadir} - y_i^{ideal}}, \tag{10}$$

y^{ideal} is found by optimizing each sub-objective independently, without considering the others, resulting in its best attainable value, and y^{nadir} is estimated from the precomputed solution pool.

5.2 User simulation and Elicitation loop

Following Defresne et al. (2025), we simulated our DMs, and their ground-truth weights are generated as $w^* = 100 \cdot \tilde{w}$, $\tilde{w} \sim \mathrm{Dirichlet}(\mathbf{1})$ for PC-TSP and the land conservation problem, $w^* \sim \mathcal{N}(25, 25/3)$ for the configuration task, and temperature $\lambda_{\text{user}} = 1$ (Defresne et al., 2025). Given a query (y_1, y_2) with $\Delta u = u(y_1, w^*) - u(y_2, w^*)$, the oracle actions are based on the following probabilities:

$$\begin{aligned} P(\text{indifference} \mid y_1, y_2) &= e^{-\lambda_{\text{user}} |\Delta u|}, \\ P(y_1 \succ y_2 \mid \neg\, \text{indifference}) &= \sigma(\lambda_{\text{user}} \Delta u)\,, \\ P(y_2 \succ y_1 \mid \neg\, \text{indifference}) &= \sigma(-\lambda_{\text{user}} \Delta u)\,. \end{aligned} \tag{11}$$

The loop ends if the absolute regret of the recommendation falls below $\ln 2/\lambda$, with the stopping rule using the deterministic threshold of $e^{-\lambda |\Delta u|} > \frac{1}{2}$ (Defresne et al., 2025). Runs that never reach this point are terminated by the query budget of 100 DM interactions and are counted as unsatisfied.

5.3 Metrics

Regret. To put different utility values into perspective, we use *regret*. Regret can be interpreted as a penalty for taking a suboptimal solution. Absolute regret $\mathrm{Regret}(y; w^*)$ and relative regret $r(y; w^*)$ are defined as (Defresne et al., 2025):

$$\begin{aligned} \mathrm{Regret}(y; w^*) &= u(y^*, w^*) - u(y, w^*) \\ r(y; w^*) &= \frac{u(y^*, w^*) - u(y, w^*)}{u(y^*, w^*)}. \end{aligned} \tag{12}$$

Unless otherwise specified, we will be using relative regret to compare different proposed solutions in the experimental section.

Representational capacity. We compared four different user model frameworks, each having a different family of posteriors they can represent, and rank them: Linear Ensemble $<$ Laplace $<$ Nonparametric(pSGLD) $<$ Normalizing Flow. We also rank each of these user models based on their end-regret. We calculate the Spearman (1904) correlation between the two rankings, and we take the median value over the 20 user seeds as a measure for *capacity*.

Reference posterior. To measure how close different approximate uncertainty models are to a Bradley & Terry (1952) posterior, we actually have to have a reference approximation of that posterior. The NUTS sampler (Hoffman & Gelman, 2014) is often used in this setting (Magnusson et al., 2025; Izmailov et al., 2021).

For each simulated user u, we fix a dataset $\mathcal{D}_u$ of 100 query pairs drawn uniformly over the solution pool and answered by the Bradley & Terry (1952) oracle. Its posterior:

$$p(w \mid \mathcal{D}_u) \;\propto\; p(w) \prod_{(a,b)\in\mathcal{D}_u} \sigma\big(\lambda\, w^\top(\phi_a - \phi_b)\big), \qquad p(w) = \mathrm{Dir}(w; \mathbf{1}), \tag{13}$$

with $\lambda = 1$ is known up to a constant factor. We sample points of that posterior using NUTS (Hoffman & Gelman, 2014), which was implemented using NumPyro (Phan et al., 2019). The sampler converged for every user and problem (split-$\hat{R} \leq 1.003$, effective sample size $\geq$3 000, no divergent transitions) (Gelman et al., 2013; Gelman & Rubin, 1992), so we treat its draws as exact samples of the posterior and collect them in the reference set $R_u = \{v_j\}_{j=1}^{M}$, $v_j \sim p(w \mid \mathcal{D}_u)$. Each user model m is fitted to the same $\mathcal{D}_u$; its approximate posterior is the set of its N_m particles, $Q_{m,u} = \{w_i\}_{i=1}^{N_m}$, projected onto the simplex. R_u and $Q_{m,u}$ enables the comparable measure of posterior fidelity over each.

Posterior fidelity (excess). We measure the distance between $Q_{m,u}$ and R_u with the energy distance (Székely & Rizzo, 2013):

$$\mathcal{E}(P, Q) = 2\,\mathbb{E}\|X - Y\| - \mathbb{E}\|X - X'\| - \mathbb{E}\|Y - Y'\|, \qquad X, X' \sim P,\; Y, Y' \sim Q, \tag{14}$$

which is zero exactly when $P = Q$. Estimated from only N_m particles, the tuned hyperparameter for user model m, is often too small (Chong & Forsyth, 2020) and gives an upward bias, as derived in Appendix Section A.3, so even exact posterior samples score above zero. We correct for this with the score of an ideal sampler holding the same number of particles: we split R_u at random into two halves, draw N_m points from one half, and score them against the other, averaging over 20 such splits. Then

$$\mathrm{excess}_{m,u} = \hat{\mathcal{E}}(R_u, Q_{m,u}) - \hat{\mathcal{E}}_{\text{exact}}(N_m), \tag{15}$$

where $\hat{\mathcal{E}}_{\text{exact}}(N_m)$ is that ideal-sampler's scoring, measuring the *excess* over a size-matches exact sampler, which may be negative. An excess of zero means the model is as close to the posterior as exact sampling with N_m particles. Closeness to the reference posterior is computed without query selection, so we measured excess in separate experiment runs.

Decision accuracy (acc). The combinatorial optimization solver receives $\bar{w}_{m,u} = \frac{1}{N_u}\sum_i w_i$, the posterior mean at the end of the elicitation loop. To measure the accuracy of how well $\bar{w}_{m,u}$ ranks two solutions from a held-out set of K pairs compared to the true weight w_u^*. We use Chu & Ghahramani (2005)'s measure for accuracy for a set of K particles:

$$\mathrm{acc}_{m,u} = \frac{1}{K}\sum_{k=1}^{K} \mathbb{I}\big[\mathrm{sign}\big(\bar{w}_{m,u}^\top \Delta\phi_k\big) = \mathrm{sign}\big(w_u^{*\top}\Delta\phi_k\big)\big], \qquad \Delta\phi_k = \phi_{a_k} - \phi_{b_k}, \tag{16}$$

with $\mathbb{I}$ the indicator function and computed on recorded particle sets from the CPE experiments.

5.4 Results

Experimental setup. For all tunings and experiments, a Hetzner `CCX63` VPS instance was selected for its 48 fast `AMD EPYC™ Milan` CPU cores. This allowed Optuna Akiba et al. (2019) to run 4 tuning steps for 12 simulated users in parallel. Even though the system had 192 GB of RAM, only about 32 GB of it was ever used. The Python version of the code was 3.13.

Q1 (representational capacity vs. end-regret). We measured the capacity, excess, and accuracy of the Ensemble, Laplace, Nonparametric, and Flow model frameworks on the four CPE instances, summarized in Table 1. A positive Spearman (1904) correlation means: *less* expressive (capacity); closeness to the exact posterior (*excess*); or a more accurate posterior mean (*acc*) goes with lower regret.

Table 1: Per-user rank correlation with end-regret over the capacity classes, median over 20 users, Wilcoxon p in brackets. Columns compare expressivity (capacity), closeness to the exact posterior (*excess*), and accuracy of the posterior mean (*acc*) when ranked against end-regret. † indicates not significant if Ensemble is dropped.

Problem	d	capacity	excess	acc
TSP-10	5	$+0.25$ (0.360)	-0.04 (0.480)	$+0.39$ (0.010)
TSP-20	5	$+0.34$ (0.401)	$+0.16$ (0.872)	$+0.27$ (0.130)
PC	77	$+0.52$ (<0.001)	$+0.36$ (<0.001)†	$+0.25$ (0.004)†
Land	4	-0.12 (0.279)	$+0.23$ (0.143)	$+0.14$ (0.411)‡

Table 2: Comparison of end-regret, number of queries when reranking based on disagreement (DRES, ours) against BALD and EVOI. Fixed ensemble size $n = 25$. Mean over 40 user seeds $\pm$ SE. *: differs from DRES. (paired Wilcoxon, Holm-corrected per metric over the table, $p < 0.05$).

	end regret			queries			sat. (%)		
Problem	DRES	BALD	EVOI	DRES	BALD	EVOI	DRES	BALD	EVOI
TSP-10	0.114	0.109	0.117	12.4	11.8	14.8	99	99	99
TSP-20	0.133	0.159	0.183*	25.6	23.6	34.0	91	93	82
PC	0.046	0.064	0.084	85.0	82.9	87.5	42	40	25

Across the four evaluated user models, there is only one significant instance (PC) where lower representational capacity correlates to end-regret. Riquelme et al. (2018) similarly found that models with lower representational power in high-dimensional settings remain competitive. Every other row indicates a positive relationship, but it is insignificant. Excess, which measures closeness to a reference posterior, does not predict end-regret development. A relationship that tracks regret better is the accuracy of the posterior mean. For all problems, the mean of user models that rank held-out query pairs better also tend to recommend better solutions.

Since the combinatorial optimization solver only receives the posterior mean $\bar{w}$, decision quality depends on the direction of $\bar{w}$, namely how the different objectives are weighted, and less so on how faithfully the user model is able to represent the spread. We will use the Ensemble model as user model for the comparisons below, since it often offers the best end-regret and number of queries tradeoff indicated in Appendix Table 5.

Q2 (impact of reranking). Since disagreement is not the only method of reranking query pairs, we evaluate how BALD (Houlsby et al., 2011b) and EVOI (Vendrov et al., 2020), both principled criteria, influence the elicitation quality in Table 2. We paired these reranking strategies on 40 user seeds for 5 independent runs over 4 CPE tasks, applying Holm (1979)-correction per metric over paired-Wilcoxon (1945).

First, BALD and EVOI require that the particles be real samples from some posterior. The ensemble of linear models does not satisfy this requirement: the initial spread comes from independent initialization, not a prior distribution. The BALD and EVOI scores themselves are calculated correctly, treating ensemble members as if they were posterior samples and using each member's Bradley & Terry (1952) preference probability. Table 2 shows that EVOI in TSP-20 is outperformed by DRES, but we attribute this mostly to a smaller candidate pool.

We expect BALD and EVOI to be more competitive with a genuine posterior approximation and a larger candidate set, so we will not claim one is better than another here. DRES will be used for further comparisons since it handles particle sets from non-Bayesian approaches.

In Table 3, we evaluate the current SOTA selection criterion by Defresne et al. (2025) (cluster) against single-pair sampling (ES) and multi-pair reranking using DRES. Each arm in this comparison was tuned independently and received 5 independent runs over the same 40 user seeds, distinct from the tuning set. We collect the end-regret and number of queries required to satisfy the DM.

Table 3: Comparison of end-regret number of queries and *sat.*, % of users satisfied after introducing single pair sampling (ES), disagreement reranking (DRES) against the cluster SOTA. *: differs from DRES (paired Wilcoxon, Holm-corrected per metric, $p < 0.05$).†Land is normalized regret.

	end regret			queries			sat. (%)		
Problem	cluster	ES	DRES	cluster	ES	DRES	cluster	ES	DRES
TSP-10	0.136*	0.121*	0.114	33.0*	20.3*	12.4	89*	96	99
TSP-20	0.415*	0.186*	0.129	68.3*	44.2*	29.5	47*	76*	91
PC	0.107*	0.095*	0.046	86.1	95.8*	85.0	28	12*	42
Land†	0.0050	0.0047	0.0057	9.9	9.6	9.1	95	100	98

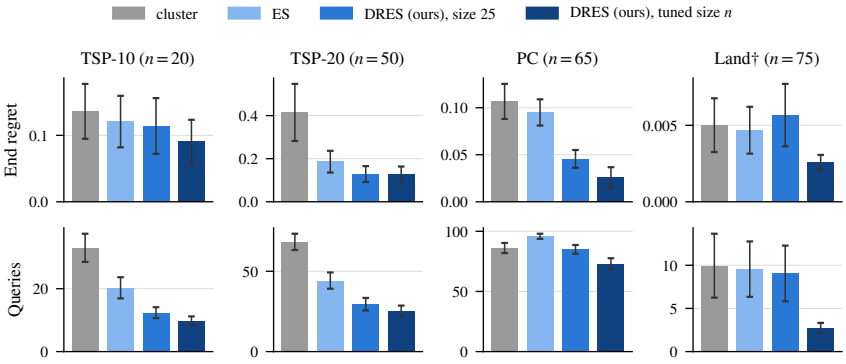

Figure 1: Comparison ladder over the mean of 40 user seeds of end-regret and number of queries. We compare the SOTA (cluster), single-sample query selection (ES), reranking multiple sampled pairs (DRES), and reranking with problem-specific ensemble sizes (DRES, tuned n). Bars show ± 1 standard error. Land† regret is normalized.

DRES obtains significantly (paired-Wilcoxon (1945), Holm (1979)-corrected) better end-regret compared to ES and cluster on TSP-10, TSP-20, and PC and often satisfies more DMs. It satisfies DMs faster than cluster and ES on TSP-10 and TSP-20. Land conservation remains flat.

Q3 (impact of ensemble size). We investigate if the elicitation performance might benefit from a problem-specific ensemble size. We do so by evaluating the difference in elicitation performance for two tuning configurations: one where the ensemble size is fixed to 25, the default configuration proposed by Defresne et al. (2025), and another where this size is allowed to be tuned. Both methods used DRES as the query selection strategy on the same 12 tuning seeds. As seen in Figure 1, across all problem instances, end-regret and the number of queries improved, indicating that ensemble size has some relationship to the combinatorial space we elicit preferences over. For a full comparison of Figure 1, we refer to Appendix Table 4.

6 Conclusion

Decision makers in multi-objective combinatorial optimization (MOCO) must choose one solution from a large set of Pareto-optimal alternatives. In CPE, each query costs user effort, so the system must ask few questions and still return a high-quality solution. We proposed a sampling-based query selection criterion that improved both end-regret and number of queries compared to the current SOTA. Different user models represent uncertainty in different ways: some keep an explicit posterior, others keep only point estimates. We evaluated how representational capacity relates to end-regret development and found that the predictive accuracy has a stronger link to decision quality than the approximate beliefs ability for uncertainty estimation.

While our proposed query selection criterion performs strong on simulated DMs, a study with real-life users could further motivate sample-based selection for real-world CPE applications.

7 AI / LLM DISCLOSURE

We used the generative AI tools Claude Opus 5 and Claude Opus 5.5 through the web interface and Claude Code for light language editing (fixing grammatical errors and suggestions for improving flow), retrieval and discovery (exploring the literature as a semantic search engine), research execution (aiding in writing code and generating scripts for exporting graphs and tables), and aid in proving one mathematical claim, (Appendix Section A.3, Claude Opus 5.5 found the plug-in V-statistic estimator of Székely & Rizzo (2013), and corrected notation like the $\otimes$ symbol).

Generative AI tools were NOT used for generating synthetic data sets, providing theoretical models or conceptual frameworks, formulating mathematical claims, proposing or refining hypotheses, designing or providing feedback on research methodology or experiments, nor did we use these tools to write sections for this work.

REFERENCES

Jamshid Aghaei, Amir Baharvandi, Mohammad-Amin Akbari, Kashem M Muttaqi, Mohammad-Reza Asban, and Alireza Heidari. Multi-objective phasor measurement unit placement in electric power networks: Integer linear programming formulation. *Electric Power Components and Systems*, 43(17):1902–1911, 2015.

Shipra Agrawal and Navin Goyal. Thompson Sampling for Contextual Bandits with Linear Payoffs, February 2014.

J. Aitchison. *The Statistical Analysis of Compositional Data*. Springer Netherlands, Dordrecht, 1986. ISBN 978-94-010-8324-9 978-94-009-4109-0. doi: 10.1007/978-94-009-4109-0.

Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2019.

Alain Billionnet. *Designing Protected Area Networks: A Mathematical Programming Approach*. EDP Sciences, 2021. ISBN 978-2-7598-2537-0.

Fritz Bökler and Petra Mutzel. Output-sensitive algorithms for enumerating the extreme nondominated points of multiobjective combinatorial optimization problems. In *Algorithms-ESA 2015: 23rd Annual European Symposium, Patras, Greece, September 14-16, 2015, Proceedings*, pp. 288–299. Springer, 2015.

Ralph Allan Bradley and Milton E. Terry. Rank Analysis of Incomplete Block Designs: I. The Method of Paired Comparisons. *Biometrika*, 39(3/4):324–345, 1952. ISSN 0006-3444. doi: 10.2307/2334029.

Darius Braziunas and Craig Boutilier. Minimax regret based elicitation of generalized additive utilities, June 2012.

Huy N. Chau and Miklos Rasonyi. Stochastic Gradient Hamiltonian Monte Carlo for Non-Convex Learning, February 2020.

Tianqi Chen, Emily B. Fox, and Carlos Guestrin. Stochastic Gradient Hamiltonian Monte Carlo. https://arxiv.org/abs/1402.4102v2, February 2014.

Chao-Kai Chiang, Takashi Ishida, and Masashi Sugiyama. LLM Routing with Dueling Feedback, October 2025.

Min Jin Chong and David Forsyth. Effectively Unbiased FID and Inception Score and where to find them, June 2020.

Paul Christiano, Jan Leike, Tom B. Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences, February 2023.

Wei Chu and Zoubin Ghahramani. Preference learning with Gaussian processes. In *Proceedings of the 22nd International Conference on Machine Learning*, ICML '05, pp. 137–144, New York, NY, USA, August 2005. Association for Computing Machinery. ISBN 978-1-59593-180-1. doi: 10.1145/1102351.1102369.

Lee Cohen, Ulrike Schmidt-Kraepelin, and Yishay Mansour. Dueling Bandits with Team Comparisons. In *Advances in Neural Information Processing Systems*, volume 34, pp. 20633–20644. Curran Associates, Inc., 2021.

Marianne Defresne, Jayanta Mandi, and Tias Guns. Preference Elicitation for Multi-objective Combinatorial Optimization with Active Learning and Maximum Likelihood Estimation, August 2025.

Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using Real NVP, February 2017.

Paolo Dragone, Stefano Teso, and Andrea Passerini. Constructive Preference Elicitation over Hybrid Combinatorial Spaces, May 2018.

Marco Foschini, Emilio Gamba, Lucas Kletzander, and Tias Guns. From cp modeling to preference elicitation in hmlv assembly problems. In *32nd International Conference on Principles and Practice of Constraint Programming (CP 2026)*, 2026a.

Marco Foschini, Dimos Tsouros, Bistra Dilkina, and Tias Guns. Anytime Lexicographic Enumeration of the Pareto Front in Multi-Objective Combinatorial Optimisation. *Journal of Multi-Criteria Decision Analysis*, 33(1):e70029, 2026b. ISSN 1099-1360. doi: 10.1002/mcda.70029.

Andrew Gelman and Donald B. Rubin. Inference from iterative simulation using multiple sequences. *Statistical Science*, 7(4):457–472, 1992.

Andrew Gelman, John B. Carlin, Hal S. Stern, David B. Dunson, Akti Vehtari, and Donald B. Rubin. *Bayesian Data Analysis*. Chapman & Hall/CRC Texts in Statistical Science Series. CRC, Boca Raton, Florida, 3 edition, 2013. ISBN 978-1-4398-4095-5 1-4398-4095-4.

Paolo Gianessi, Audrey Cerqueus, Damien Lamy, and Xavier Delorme. Using reconfigurable manufacturing systems to minimize energy cost: a two-phase algorithm. *IFAC-PapersOnLine*, 54(1): 379–384, 2021.

Shengbo Guo and Scott Sanner. Multiattribute Bayesian Preference Elicitation with Pairwise Comparison Queries. In Liqing Zhang, Bao-Liang Lu, and James Kwok (eds.), *Advances in Neural Networks - ISNN 2010*, pp. 396–403, Berlin, Heidelberg, 2010. Springer. ISBN 978-3-642-13278-0. doi: 10.1007/978-3-642-13278-0_51.

Matthew D. Hoffman and Andrew Gelman. The no-u-turn sampler: Adaptively setting path lengths in hamiltonian monte carlo. *Journal of Machine Learning Research*, 15(1):1593–1623, 2014. doi: 10.5555/2627435.2638586. URL https://jmlr.org/papers/v15/hoffman14a.html.

Sture Holm. A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*, 6(2):65–70, 1979.

Neil Houlsby, Ferenc Huszár, Zoubin Ghahramani, and Máté Lengyel. Bayesian Active Learning for Classification and Preference Learning, December 2011a.

Neil Houlsby, Ferenc Huszár, Zoubin Ghahramani, and Máté Lengyel. Bayesian Active Learning for Classification and Preference Learning. https://arxiv.org/abs/1112.5745v1, December 2011b.

Tian Huang, Shengbo Wang, and Ke Li. Direct Preference-Based Evolutionary Multi-Objective Optimization with Dueling Bandits. In *Advances in Neural Information Processing Systems*, volume 37, pp. 122206–122258. Curran Associates, Inc., 2024. doi: 10.52202/079017-3884.

Pavel Izmailov, Sharad Vikram, Matthew D. Hoffman, and Andrew Gordon Wilson. What are Bayesian neural network posteriors really like? In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, 2021.

Ralph L. Keeney and Howard Raiffa. Tradeoffs under certainty. In *Decisions with Multiple Objectives: Preferences and Value Trade-Offs*, pp. 66–130. Cambridge University Press, Cambridge, 1993a.

Ralph L. Keeney and Howard Raiffa. Multiattribute preferences under uncertainty: More than two attributes. In *Decisions with Multiple Objectives: Preferences and Value Trade-Offs*, pp. 282–353. Cambridge University Press, Cambridge, 1993b.

Kimin Lee, Laura Smith, and Pieter Abbeel. PEBBLE: Feedback-Efficient Interactive Reinforcement Learning via Relabeling Experience and Unsupervised Pre-training, June 2021.

Chunyuan Li, Changyou Chen, David Carlson, and Lawrence Carin. Preconditioned Stochastic Gradient Langevin Dynamics for Deep Neural Networks. https://arxiv.org/abs/1512.07666v1, December 2015.

D. V. Lindley. On a Measure of the Information Provided by an Experiment. *The Annals of Mathematical Statistics*, 27(4):986–1005, December 1956. ISSN 0003-4851, 2168-8990. doi: 10.1214/aoms/1177728069.

Chang Liu and Jun Zhu. Riemannian Stein Variational Gradient Descent for Bayesian Inference. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1), April 2018. ISSN 2374-3468. doi: 10.1609/aaai.v32i1.11810.

Dong C Liu and Jorge Nocedal. On the limited memory BFGS method for large scale optimization. *Mathematical Programming*, 45(1-3):503–528, 1989.

Qiang Liu and Dilin Wang. Stein Variational Gradient Descent: A General Purpose Bayesian Inference Algorithm, September 2019.

Xiuyuan Lu and Benjamin Van Roy. Ensemble Sampling. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

Måns Magnusson, Jakob Torgander, Paul-Christian Bürkner, Lu Zhang, Bob Carpenter, and Aki Vehtari. Posteriordb: Testing, benchmarking and developing Bayesian inference algorithms. In *Proceedings of the 28th International Conference on Artificial Intelligence and Statistics*, volume 258 of *Proceedings of Machine Learning Research*, 2025.

Lorraine McGinty and Barry Smyth. Tweaking critiquing. In *Proceedings of the Workshop on Personalization and Web Techniques at the International Joint Conference on Artificial Intelligence (IJCAI-03)*, pp. 20–27. Morgan Kaufmann San Francisco, 2003.

Mariana Mesquita-Cunha, José Rui Figueira, and Ana Paula Barbosa-Póvoa. New ε- constraint methods for multi-objective integer linear programming: A pareto front representation approach. *European Journal of Operational Research*, 306(1):286–307, 2023.

Kaisa Miettinen. A priori methods. In *Nonlinear Multiobjective Optimization*, pp. 115–129. Springer, 1998a.

Kaisa Miettinen. *Nonlinear Multiobjective Optimization*, volume 12 of *International Series in Operations Research & Management Science*. Springer US, Boston, MA, 1998b. ISBN 978-1-4613-7544-9 978-1-4615-5563-6. doi: 10.1007/978-1-4615-5563-6.

Petrus Mikkola, Luigi Acerbi, and Arto Klami. Preferential Normalizing Flows, October 2024.

Jorge Nocedal and Stephen J Wright. *Numerical Optimization*. Springer, 2 edition, 2006.

Ellen R. Novoseller, Yibing Wei, Yanan Sui, Yisong Yue, and Joel W. Burdick. Dueling Posterior Sampling for Preference-Based Reinforcement Learning. https://arxiv.org/abs/1908.01289v4, August 2019.

Du Phan, Neeraj Pradhan, and Martin Jankowiak. Composable Effects for Flexible and Accelerated Probabilistic Programming in NumPyro, December 2019.

Chao Qin, Zheng Wen, Xiuyuan Lu, and Benjamin Van Roy. An Analysis of Ensemble Sampling, March 2023.

Carlos Riquelme, George Tucker, and Jasper Snoek. Deep bayesian bandits showdown: An empirical comparison of bayesian deep networks for thompson sampling, 2018.

Daniel Russo and Benjamin Van Roy. An Information-Theoretic Analysis of Thompson Sampling. *Journal of Machine Learning Research*, 17(68):1–30, 2016. ISSN 1533-7928.

Daniel Russo, Benjamin Van Roy, Abbas Kazerouni, Ian Osband, and Zheng Wen. A Tutorial on Thompson Sampling, July 2020.

Aadirupa Saha and Aditya Gopalan. Combinatorial Bandits with Relative Feedback. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

H. S. Seung, View Profile, M. Opper, View Profile, H. Sompolinsky, and View Profile. Query by committee. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, ACM Conferences, pp. 287–294. Association for Computing Machinery, July 1992. ISBN 978-0-89791-497-0. doi: 10.1145/130385.130417.

Charles Spearman. The proof and measurement of association between two things. *The American Journal of Psychology*, 15(1):72–101, 1904.

Yanan Sui, Vincent Zhuang, Joel W. Burdick, and Yisong Yue. Multi-dueling Bandits with Dependent Arms, April 2017.

Gábor J. Székely and Maria L. Rizzo. Energy statistics: A class of statistics based on distances. *Journal of Statistical Planning and Inference*, 143(8):1249–1272, August 2013. ISSN 0378-3758. doi: 10.1016/j.jspi.2013.03.018.

Stefano Teso, Andrea Passerini, and Paolo Viappiani. Constructive Preference Elicitation by Setwise Max-margin Learning, April 2016.

W. R. Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25:285–294, 1933. doi: 10.1093/biomet/25.3-4.285.

Bastián Véjar, Gaël Aglin, Ali İrfan Mahmutoğulları, Siegfried Nijssen, Pierre Schaus, and Tias Guns. An efficient structured perceptron for np-hard combinatorial optimization problems. In *International Conference on the Integration of Constraint Programming, Artificial Intelligence, and Operations Research*, pp. 253–262. Springer, 2024.

Ivan Vendrov, Tyler Lu, Qingqing Huang, and Craig Boutilier. Gradient-Based Optimization for Bayesian Preference Elicitation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(06):10292–10301, April 2020. ISSN 2374-3468, 2159-5399. doi: 10.1609/aaai.v34i06.6592.

Paolo Viappiani and Craig Boutilier. Optimal Bayesian recommendation sets and myopically optimal choice query sets. In *Proceedings of the 24th International Conference on Neural Information Processing Systems - Volume 2*, volume 2 of *NIPS'10*, pp. 2352–2360, Red Hook, NY, USA, December 2010. Curran Associates Inc.

Siwei Wang and Wei Chen. Thompson sampling for combinatorial semi-bandits, 2022.

Thomas A. Weber. Relatively robust decisions. *Theory and Decision*, 94(1):35–62, January 2023. ISSN 1573-7187. doi: 10.1007/s11238-022-09866-z.

Max Welling and Yee Whye Teh. Bayesian Learning via Stochastic Gradient Langevin Dynamics. *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pp. 681–688, 2011. doi: 10.5555/3104482.3104568.

Frank Wilcoxon. Individual comparisons by ranking methods. *Biometrics Bulletin*, 1(6):80–83, 1945. ISSN 00994987. URL `http://www.jstor.org/stable/3001968`.

Huasen Wu and Xin Liu. Double Thompson Sampling for Dueling Bandits. In *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016.

Yisong Yue and Thorsten Joachims. Beat the mean bandit. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, ICML'11, pp. 241–248, Madison, WI, USA, June 2011. Omnipress. ISBN 978-1-4503-0619-5.

Yisong Yue, Josef Broder, Robert Kleinberg, and Thorsten Joachims. The *K*-armed dueling bandits problem. *Journal of Computer and System Sciences*, 78(5):1538–1556, September 2012. ISSN 0022-0000. doi: 10.1016/j.jcss.2011.12.028.

Xinyu Zhang and Cheng Li. Pigeonhole Stochastic Gradient Langevin Dynamics for Large Crossed Mixed Effects Models, May 2025.

Masrour Zoghi, Shimon Whiteson, Remi Munos, and Maarten Rijke. Relative upper confidence bound for the k-armed dueling bandit problem. In Eric P. Xing and Tony Jebara (eds.), *Proceedings of the 31st International Conference on Machine Learning*, Proceedings of Machine Learning Research, pp. 10–18, Beijing, China, 22–24 Jun 2014. PMLR. URL https://proceedings.mlr.press/v32/zoghi14.html.

A Appendix

A.1 Transporting weight-space to ALR

preliminaries. The negative log-posterior in ALR coordinates is:

$$-\log p(u \mid \mathcal{D}) \;\propto\; \underbrace{\sum_{t=1}^{N} \operatorname{softplus}\big(-\lambda\, w(u)\cdot(\phi_t^+ - \phi_t^-)\big)}_{\text{BT NLL}} \;+\; \underbrace{\frac{1}{2\sigma_0^2}\|u\|^2}_{\text{Gaussian prior in ALR}}$$

with $\mathcal{D} = \{(\phi_t^+, \phi_t^-)\}_{t=1}^N$ the set of all observed comparisons and λ the Bradley-Terry inverse temperature.

Let $g^w = \nabla_w \log p$ be the gradient in weight space, which is computed by all of the original algorithms. The Jacobian for the softmax-with-zero[1] is:

$$J_{ji}(u) = \frac{\partial w_j}{\partial u_i} = w_j\big(\mathbb{I}[j=i] - w_i\big)$$

for $i \in \{1, \dots, D-1\}$ and $j \in \{1, \dots, D\}$. When we push g^w through, it gives us the gradient in ALR coordinates in closed form:

$$g_i^{alr} = \sum_{j=1}^{D} J_{ji}(u)\, g_j^w \;=\; w_i\Big(g_i^w - \sum_{k=1}^{D} w_k\, g_k^w\Big), \qquad i = 1, \dots, D-1.$$

g^{alr} is a drop-in replacement for the gradient the methods would already use.

A.2 Application of ALR to uncertainty models

Laplace approximation. The MAP estimate $u^* = \arg\min_u -\log p(u \mid \mathcal{D})$ is found via the *L-BFGS* algorithm Liu & Nocedal (1989) with a strong Wolfe line search Nocedal & Wright (2006). The precision matrix Λ, the Hessian of $-\log p(u|\mathcal{D})$, is obtained by automatic differentiation. In order to ensure that $\tilde{\Lambda}$ is positive definite in all directions, especially the directions that have not yet been included in the query responses, Λ is dampened to $\tilde{\Lambda} = \frac{1}{2}(\Lambda + \Lambda^\top) + \tau\mathbb{I}$, with $\frac{1}{2}(\Lambda + \Lambda^\top)$ to symmetrize for numerical stability. Posterior samples are drawn as $u^{(k)} = u^* + L_\Sigma z^{(k)}$, with $z^{(k)} \sim \mathcal{N}(0, \mathbb{I})$ and $L_\Sigma = \operatorname{chol}(\tilde{\Lambda}^{-1})$. When mapping $u^{(k)}$ through $w(\cdot)$, we receive the correct simplex-constrained weights to give to the ensemble model.

SGLD, pSGLD and SGHMC. These particle methods apply their published update rules to u instead of w, utilizing g^{alr} when necessary. *SGLD* takes $u \leftarrow u + \alpha(g^{alr} + \varepsilon)$ with $\varepsilon \sim \mathcal{N}(0, \sigma_{noise}^2\mathbb{I})$ Welling & Teh (2011). *pSGLD* uses g^{alr} for its diagonal preconditioning matrix, scaling both drift and injected noise Li et al. (2015). *SGHMC* maintains a velocity buffer $v \in \mathbb{R}^{D-1}$ in ALR-space and updates $v \leftarrow (1-\gamma)v + \alpha g^{alr} + \sqrt{2\gamma\alpha}\, z$, followed by $u \leftarrow u + v$ Chen et al. (2014).

[1] We have to include zero since the ALR-transform of the reference weight has to be defined as well $\ln(w_D/w_D) = 0$. This is required due to the definition of $w(\cdot)$. The $\mathbb{I}[j = i]$ indicator handles both diagonal and non-diagonal cases.

SVGD. The Stein force is computed entirely in u-space: the RBF kernel and mean bandwidth heuristic are evaluated on the ALR coordinates, and the driving term uses g^{alr} Liu & Wang (2019).

Preferential normalizing flow. The flow model models $q_\varphi(u)$, a stack of affine coupling layers (Dinh et al., 2017) over the ALR coordinates and uses $\mathcal{N}(0, \sigma_{base}^2 \mathbb{I})$ as an initial density (Mikkola et al., 2024). It is trained per round by maximizing the evidence lower bound, using the same Bradley-Terry likelihood on $w(u)$ and the same Gaussian prior on u as the models above.

A.3 Finite-sample bias of the energy distance

Energy distance may be biased by the number of samples. Simply fixing the number of particles to compare between different uncertainty models is insufficient (Chong & Forsyth, 2020). We investigate this more formally below.

Let $X = \{x_i\}_{i=1}^{L}$ be independent and identically distributed (i.i.d.) draws from P and $Y = \{y_j\}_{j=1}^{N}$ be i.i.d. drawn from Q and independent from X. Let $\delta_P = \mathbb{E}\|X - X'\|$ and $\delta_Q = \mathbb{E}\|Y - Y'\|$, both finite because samples live on the simplex. We estimate $\mathcal{E}(P, Q)$ with the V-statistic estimator of Székely & Rizzo (2013):

$$\hat{\mathcal{E}}(X, Y) = \frac{2}{LN} \sum_{i=1}^{L} \sum_{j=1}^{N} \|x_i - y_j\| - \frac{1}{L^2} \sum_{i=1}^{L} \sum_{i'=1}^{L} \|x_i - x_{i'}\| - \frac{1}{N^2} \sum_{j=1}^{N} \sum_{j'=1}^{N} \|y_j - y_{j'}\|. \tag{17}$$

Lemma. $\mathbb{E}\big[\hat{\mathcal{E}}(X, Y)\big] = \mathcal{E}(P, Q) + \frac{\delta_P}{L} + \frac{\delta_Q}{N}$.

Proof. Each cross pair (x_i, y_j) is a draw from $P \otimes Q$, so the first term has expectation $2\,\mathbb{E}\|X - Y\|$. In the second term the L diagonal entries are zero and each of the $L(L-1)$ off-diagonal entries has expectation δ_P, so the term has expectation $\frac{L-1}{L}\delta_P = \delta_P - \delta_P/L$. The third term likewise has expectation $\delta_Q - \delta_Q/N$. Substituting into the definition of $\mathcal{E}$ gives the result. □

Even exact samples, when Q equals P, score $\delta_P(1/L + 1/N) > 0$ in expectation.

Consequence for the excess metric. A user model m is scored with L reference draws and its N_m particles, and the ideal sampler with L draws from one half of R_u and N_m draws from the other half. By the lemma,

$$\mathbb{E}\big[\hat{\mathcal{E}}(R_u, Q_{m,u})\big] = \mathcal{E}(P, Q_m) + \frac{\delta_P}{L} + \frac{\delta_{Q_m}}{N_m}, \tag{18}$$

$$\mathbb{E}\big[\hat{\mathcal{E}}_{\text{exact}}(N_m)\big] = \frac{\delta_P}{L} + \frac{\delta_P}{N_m}, \tag{19}$$

$$\mathbb{E}\big[\text{excess}_{m,u}\big] = \mathcal{E}(P, Q_m) + \frac{\delta_{Q_m} - \delta_P}{N_m}. \tag{20}$$

The reference-side bias cancels out exactly. The particle-side bias cancels only when the model's particles are as spread out as the posterior ($\delta_{Q_m} = \delta_P$). A model with under-dispersed particles has its excess underestimated by at most δ_P/N_m, and an over-dispersed model has it overestimated. Averaging over 20 splits reduces the variance of the floor, not this bias.

A.4 DRES implementation details

Clipping. The implementation of line 5 of the algorithm is:

$$p_m \leftarrow \text{clip}\left(\sigma((U_{y_1,m} - U_{y_2,m})/\tau),\, \varepsilon,\, 1 - \varepsilon\right).$$

This is important for BALD reranking. We clip using $\varepsilon = 10^{-3}$ to prevent $0 \log 0$ during the entropy calculations. We do not clip for EVOI reranking, since a zero predicted probability is assigned zero weight.

Hyperparameters. The number of sampled query pairs $K = 8$, total attempts to sample at most K distinct query pairs $R = 50$ and $\tau = 1.0$ is the scoring temperature.

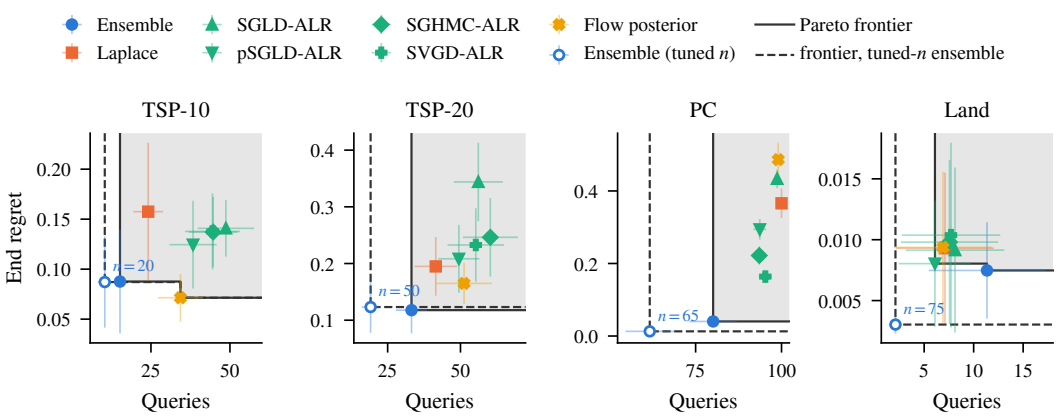

Figure 2: The Pareto frontier over end-regret and number of queries for eight different user model over four CPE tasks

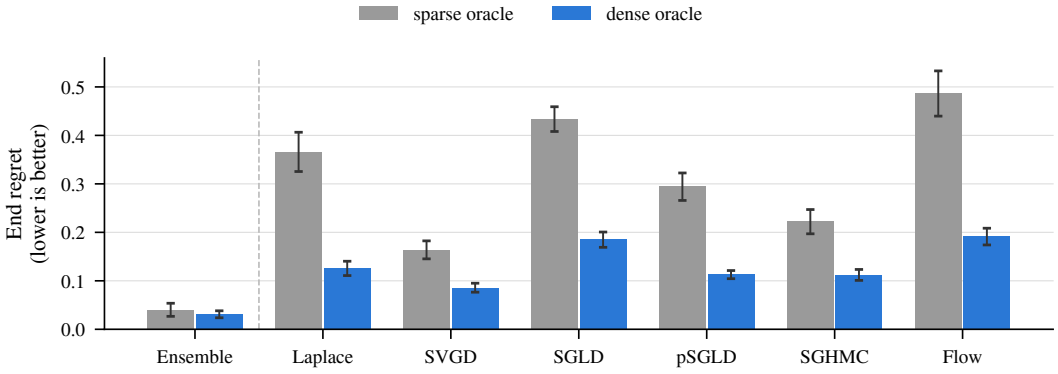

Figure 3: Oracle ablation test on the configuration CPE task

Fallbacks. If $|\mathbf{B}| = 0$, all particles share one favorite solution. Since this collapse only occurs after user satisfaction or invalid initialization, we sample two solutions uniformly over the solution pool. $|\mathbf{B}| = 1$ or $M < 3$ equals ensemble sampling, so we skip the reranking calculations. If this fails after R attempts, we sample two solutions uniformly.

A.5 Plots

PC oracle ablation The computer configuration (PC) task defaults to sparse users, where about 80% of w^* are zeroed out. Since the ALR parametrization is not able to handle the edges of the simplex, we disable this setting to investigate if the ranking over models changes. Figure 3 indicates that the relative performance between ALR-parametrized methods remains the same. Furthermore, their performance significantly improves against sparse oracle weights.

A.6 Tables

A.7 RQ1 model selection

Runtime. Every user model-query selection pair replays the same recorded 100-query trajectory per user, so at each step all models have seen the same data. Per user: minimum over 10 timed replays, averaged over queries 1, 10, 25, 50, 75 and 100; mean over users. 95% confidence intervals are within 6% of the mean for the update and within 26% for selection. Single-threaded, one worker per physical core, so one worker per hyperthreading group; timings under full load are within 11%

Table 4: Effect on ensemble size for DRES, 40 user seeds, mean over users. *: differs from default size 25 (paired Wilcoxon, Holm-corrected per metric, $p < 0.05$). Within-row parameters are fixed

		end regret		queries	
Problem	n	25	n	25	n
TSP-10	20	0.106	0.091*	11.6	9.8
TSP-20	50	0.138	0.127	22.6	25.2
PC	65	0.054	0.026*	80.6	73.0
Land†	75	0.0057	0.0026*	9.1	2.8*

Table 5: The seven user models under DRES (the RQ1 runs), same 20 user seeds and runs as RQ1, mean over users; bold: lowest regret per problem. *: differs from the ensemble (paired Wilcoxon, Holm-corrected over the six comparisons per problem and metric, $p < 0.05$). Land regret is normalised.

	end regret				queries			
Model	TSP-10	TSP-20	PC	Land	TSP-10	TSP-20	PC	Land
Ensemble	0.088	**0.118**	**0.040**	**0.0075**	15.2	33.2	80.1	11.3
Laplace	0.157	0.195*	0.366*	0.0093	24.1	41.5	100.0	7.1
SGLD-ALR	0.141	0.344	0.434*	0.0091	48.7*	56.1	98.7	8.1
pSGLD-ALR	0.124	0.208	0.294*	0.0080	38.3*	49.4	93.7	6.1
SGHMC-ALR	0.137	0.246	0.222*	0.0098	44.7*	60.2	93.5	7.6
SVGD-ALR	0.138	0.233	0.164*	0.0104	44.5*	55.3	95.3	7.7
Flow posterior	**0.071**	0.165	0.486*	0.0094	34.3*	51.2	99.0	6.9

of the same jobs run alone. Table 6 shows the per-query wall-clock cost in milliseconds (ms) for query selection (sel.) and user model update (upd.).

Table 6: Per-query wall-clock cost (ms) of query selection (sel.) and the model update (upd.) for every user model and selection rule, 10 users.

		TSP-10		TSP-20		PC		Land	
Model	Rule	sel.	upd.	sel.	upd.	sel.	upd.	sel.	upd.
Ensemble	cluster	15.7	1.89	15.2	1.87	6.94	2.15	7.67	1.76
	ES	1.97	1.78	2.45	1.78	0.81	1.77	0.58	1.76
	DRES	7.06	1.71	9.56	1.82	7.58	2.78	5.20	1.85
Laplace	ES	2.19	3.86	3.27	4.39	6.28	12.1	0.34	3.53
	DRES	2.92	3.88	3.65	4.43	10.1	12.2	2.36	3.56
SGLD-ALR	ES	1.73	8.71	1.84	34.9	1.52	211	0.13	6.42
	DRES	3.48	8.70	3.58	34.9	3.27	211	2.35	6.41
pSGLD-ALR	ES	1.54	10.4	1.67	5.91	3.49	1823	0.13	16.7
	DRES	3.18	10.4	2.11	5.91	5.56	1820	2.35	16.7
SGHMC-ALR	ES	1.54	5.53	1.73	34.7	3.10	828	0.12	9.57
	DRES	1.89	5.53	1.94	34.7	5.49	828	2.35	9.59
SVGD-ALR	ES	1.57	23.2	1.67	24.3	2.21	1372	0.16	20.3
	DRES	1.81	23.2	1.88	24.3	3.87	1373	2.35	20.3
Flow posterior	ES	2.94	4329	3.96	1177	2.04	1045	0.97	527
	DRES	3.33	4349	4.42	1179	3.06	1039	2.24	529