Multi-objective ranking for job marketplace optimization

Rong Liu[1], **Eran Brill**[1], **Ethan Barker**[1], **Ashley Chang**[1], **Yiftach Dayan**[1], **Yu Sun**[1]

[1]ZipRecruiter, Inc
604 Arizona Ave
Santa Monica, California 90401 USA
{rongl, eranb, ethanb, ashleyc, yiftachd, yus}@ziprecruiter.com

Abstract

A robust multi-objective ranking approach is developed for identifying the ranking function that arrives at the desired balance of multiple business objectives in job marketplaces. The multi-objective optimization problem is converted into a constrained optimization of a single objective, making it easy to configure/adjust for different business needs. The formulated constrained optimization is solved using a direct search algorithm that combines Augmented Lagrangian Multiplier method and Powell's method with an unbiased estimate of position bias. The approach demonstrates good robustness in a numerically ill-conditioned experimental optimization problem that stems from real business needs in job marketplaces. A further online test using the identified ranking function shows a good agreement between the online metrics values and offline simulation results.

Introduction

Job marketplace are dynamic platforms that connect employers with potential employees, or vice versa. They surface opportunities for job seekers to find employment while enabling employers to access talents that meet their requirements. Online job marketplaces have revolutionized the hiring process by broadening reach, improving efficiency, and providing valuable data to optimize recruitment efforts. It is now much easier for job seekers to explore career opportunities and for employers to identify and reach skilled candidates on job marketplaces. Job marketplaces are highly competitive and diverse, with a range of industries, job types, and employment models (full-time, part-time, freelance) and work mode (in-person, remote, travel, and hybrid). Advanced technology, such as AI-driven matching, automated screening processes, virtual interviews, and talent databases, is now playing a central role in this ecosystem.

In order to enhance the efficiency, effectiveness, and overall performance, optimization of the job marketplace is aiming to maximize value for all participants - employers, job seekers, and service providers. Major optimization objectives include: personalized matching of job seekers to job opportunities, enhancing user engagement to keep job seekers active on the platform, establishing optimal pricing for job listings and ensuring fair costs for employers while sustaining the platform's competitiveness and profitability.

Among these objectives, the balance between the best matching and fair costs is particularly important. The balance is typically achieved through an optimal ranking of the job listings. Like other e-commerce or service marketplaces, job marketplace optimization often uses two-step ranking to maintain the quality of results to job seekers while balancing efficiency and scalability (Bi et al. 2020). In the first step, a vast pool of job listings is quickly filtered down to a manageable subset of high relevance to job seekers. The subset is then re-ranked in a second step according to the desired trade-off between relevance and other business objectives, such as maximizing revenue, enhancing user engagement, balancing short term gains and long term growth, and maintaining equilibrium between supply and demand. These objectives are typically not congruent with each other and thus there is unlikely a single optimal solution that can satisfy them simultaneously. In order to address this challenge, Multi-objective optimization (MOO) techniques are widely used in job marketplace optimization (Wang et al. 2012).

Related works

Due to its crucial role in marketplace optimization, MOO has become a highly active research field and an extensive number of MOO techniques have been developed in recent years. There are some comprehensive reviews of the latest advancements and the current state of MOO techniques published in recent survey studies (Zaizi, Qassimi, and Rakrak 2023; Emmerich and Deutz 2018). MOO techniques, such as Pareto optimization (e.g., weighted sum optimization, constraint-based optimization, or hybrid optimization), evolutionary algorithms (Hua et al. 2021), and swarm intelligence (Sharifi et al. 2021), were successfully applied to solve the MOO problems for marketplace optimization.

Among these approaches, Pareto optimization with linear ranking function has the lowest computational complexity (Li et al. 2023). One of the representative works in this field (Bai, Xie, and Wang 2018) is converting the original MOO problem into a linear programming formulation (with constraints), which is then solved using Augmented Lagrangian Multiplier (ALM) method. A highlight of the work is its offline simulation framework through replaying the auctions recorded from real online ranking requests,

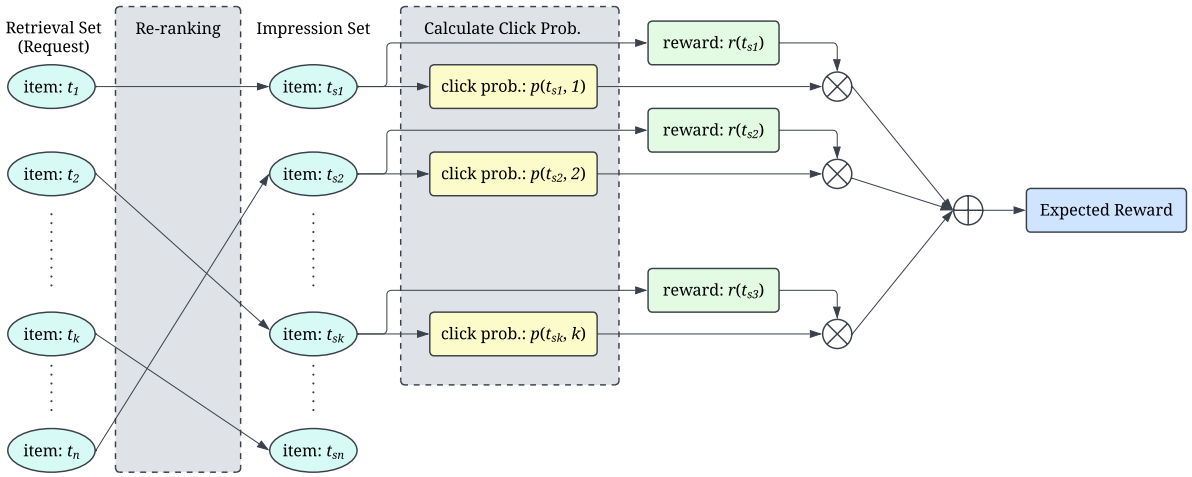

Figure 1: Auction simulation on a single request. A request retrieved based on relevance is re-ranked by a trade-off between click probability and rewards of click (e.g., bid). The reward on item t_{s_i} is provided along with the request, while the click probability has to be estimated for its new rank position i.

which is now widely used by online marketplaces. However, the over-parameterized objective function makes the optimization configuration very complex and difficult to update to keep up with rapid change of business needs.

Preliminary about offline auction simulation

In a typical online auction process, a small subset is retrieved (via search engine) from a vast pool of candidates (e.g., job listings) according to their relevance to the user. The retrieval set (also called a request) is then ranked according to marketplace policy that is usually a function of the relevance, bid, and other features. The top K items after the ranking are selected as the impression set for a placement, where K is the number of available impression positions. The performance of the ranker is evaluated by a set of metrics, such as clicks and revenue. This online process for evaluating different marketplace rankers could be very time consuming and expensive. Therefore, we usually resort to offline simulations to identify a small set of ranker variants that satisfy different business needs. The variants are then tested online for further evaluation.

For offline auction simulation (Figure 1), a data set that logs click information for a holdout of not ranked retrieval sets (i.e., these sets are sorted by relevance only) is used for the evaluation of different rankers (Bai, Xie, and Wang 2018). In the simulation a request that contains a list of items $\{t_1, t_2, \ldots, t_n\}$ is ranked by a marketplace ranker to a new ranking $\{t_{s_1}, t_{s_2}, \ldots, t_{s_n}\}$, where s denotes a ranking/permutation of $\{1, 2, \ldots, n\}$. Reward metrics r (e.g., clicks and revenue) are then calculated on the top K items from the new ranking. As the items are shuffled from their original ranks (for which we know they are clicked or not from logging), we will need to estimate the click probabilities of these items $p(t_{s_i}, i)$ on their new ranking position i. Once we have the click probabilities we can calculate the expected (estimated) value of metric r as $E(R) = \Sigma_i p(t_{s_i}, i) \cdot r(t_{s_i})$. For simulations on a group of requests, the sum of $E(R)$ over each request will be the expected metric value of the entire group.

Method

Position bias estimation

The retrieval set returned by search engines is usually selected based on a relevance score predicted by machine learning models, like Learning-to-Rank (Goswami, Zhai, and Mohapatra 2018). The relevance score is independent of position, generated by methods such as Inverse Propensity Weighting (IPW) (Bai, Xie, and Wang 2018) or Position-Aware Learning (Dai et al. 2020) to overcome position bias inherent from click data (used as training data). With position bias ρ provided along with the relevance score γ, we can use IPW approach to calculate the click probability of item t_{s_i} on its new rank position as: $p(t_{s_i}, i) = \gamma(t_{s_i}) \cdot \rho(i)$.

For fast-changing marketplaces, retrieval models used for searching will need frequent refreshing to generate relevance scores and position bias estimates that match the actual number of clicks over time. However, this full model refresh can be very expensive. A less expensive solution is to calibrate the click probability by updating position bias only. For a rank position i, the expected number of clicks is $\Sigma p(t_{s_i}, i) = \Sigma \gamma(t_{s_i}) \cdot \rho(i) = \rho(i) \Sigma \gamma(t_{s_i})$, where the sum is over all items placed at rank position i among all requests. Given the actual total number of clicks for position i is n_i, an unbiased estimate of the position bias will be $\rho(i) = n_i / \Sigma \gamma(t_{s_i})$. This estimate, as remarked below, is also an approximate Maximum Likelihood Estimate (MLE).

Remark 1: *The unbiased estimate obtained as the ratio between number of clicks and the sum of relevance score is an approximate MLE of position bias.*

Proof: We can reformulate the click probability for a given position as a coin toss problem: tossing n different coins (not a coin for n times) that have different probabilities to

get heads ($p_i(H)$ for i-th coin) one time. The probability distribution for the total number of heads follows Poisson binomial distribution (Tang and Tang 2023). Although Poisson binomial distribution is quite complex, it can be well approximated by a single Poisson distribution ($\lambda^k e^{-\lambda}/k!$, where $\lambda = \Sigma_i p_i$) when $n \gg 1$ and $p \ll 1$ (Tang and Tang 2023). For Poisson distribution the MLE of λ is: $\lambda^* = \Sigma_i k_i/N$. As we just toss the n coins one time without repeating, the sample size of the approximating Poisson distribution is $N = 1$ and thus λ^* equals the total number heads we get for the one trial (i.e., n_k). In the context of position bias estimation, there are a large number of items placed at the same rank position (i.e., $n \gg 1$) and their click probabilities are usually low (i.e., $p \ll 1$). Therefore, we will have an approximate MLE of position bias through $\rho(k)\Sigma\gamma(t_{s_k}) = n_k \approx \lambda^*$.

MOO problem formulation

With the simulation framework defined above, we can formulate a MOO problem to find the best ranking to satisfy a range of business needs. In general, objective functions are selected from reward functions, which can be re-written on the basis of position bias as: $E(R_{obj}) = \Sigma_{i\le k}\rho(t_{s_i}, i)\cdot r(t_{s_i}) = \Sigma_{i\le k}\rho(i)\cdot[\gamma(t_{s_i})\cdot r_{obj}(t_{s_i})]$. Some commonly used objectives include clicks (for which $r(\cdot) = 1$) and revenue (for which $r(\cdot)$ is the bid for an item).

In order to find the best ranking that maximizes a single objective $E(R_{obj})$, we can formulate the following optimization problem:

$$Max_{s\in S}\ \Sigma_{i\le k}\rho(i)\cdot[\gamma(t_{s_i})\cdot r_{obj}(t_{s_i})]$$

where S denotes all possible rankings (permutations) of items in a request. For most impression placements, we can assume that the position bias decreases monotonically with decreased rank (as users are inclined towards clicking on higher ranked results). Based on this assumption, the optimal solution s^* maximizing $E(R_{obj})$ will be the ranking that sorts the items in descending order of $\gamma(t)\cdot r_{obj}(t)$ (here the index is ignored as γ and r are the property of an item t and independent of sort position). For example, the ranking that yields maximum clicks will be the ranking that sorts items in descending order of relevance score γ. For maximum revenue we can rank the items by the product of their relevance score and bid.

For multiple objectives, we need to formulate a MOO problem like:

$$Max_{s\in S}\ (\Sigma_{i\le k}\rho(i)\cdot[\gamma(t_{s_i})\cdot r_{click}(t_{s_i})],\ \Sigma_{i\le k}\rho(i)\cdot[\gamma(t_{s_i})\cdot r_{bid}(t_{s_i})])$$

There is typically no feasible solution that minimizes all objective functions simultaneously (the objectives can conflict with each other). Therefore, attention is paid to find solutions that cannot be improved in any of the objectives without degrading at least one of the other objectives, i.e., Pareto optimal solutions. The set of Pareto optimal solutions is called the Pareto front.

A simple approach to find Pareto solutions for a MOO problem is linear scalarization that integrates all objectives into a single one via weighted sum:

$$Max_{s\in S}\ \Sigma_{obj\in O}\Sigma_{i\le k} w_{obj}\cdot\rho(i)\cdot[\gamma(t_{s_i})\cdot r_{obj}(t_{s_i})]$$

The integrated objective function can be re-organized as:

$$Max_{s\in S}\ \Sigma_{i\le k}\rho(i)\cdot\Sigma_{obj\in O} w_{obj}\cdot[\gamma(t_{s_i})\cdot r_{obj}(t_{s_i})]$$

For this single objective optimization problem the best solution s^* will be a ranking that sorts items in each request in descending order by a linear ranking score: $z = \Sigma_{obj\in O} w_{obj}\cdot\gamma(t)\cdot r_{obj}(t)$. This method, however, can only find the points on the convex hull of the objective set (Emmerich and Deutz 2018). In addition, as a priori method, this method requires sufficient preference information to define suitable weights. Instead of defining suitable weights, it might be easier to express some objectives as constraints while optimizing the others.

$$Max_{s\in S}\ \Sigma_{i\le k}\rho(i)\cdot[\gamma(t_{s_i})\cdot r_{obj'}(t_{s_i})]$$
$$s.t.\ \Sigma_{i\le k}\rho(i)\cdot[\gamma(t_{s_i})\cdot r_{obj}(t_{s_i})] \ge \epsilon_{obj}, obj \in O\backslash obj'$$

This method, known as the ϵ-constraint method (Emmerich and Deutz 2018), can explore both convex and non-convex Pareto fronts. The search space of this problem is huge (for n items the number of all possible permutations is $n!$). In order to reduce the complexity, we keep using the linear ranking score: $z = \Sigma_{obj\in O} w_{obj}\cdot\gamma(t)\cdot r_{obj}(t)$. As noted in the linear scalarization approach, the use of linear ranker will limit the solution to the convex hull of the objective set.

With the use of linear ranking score the assumption of monotonicity about position bias becomes unnecessary:

Remark 2: *If position bias is not monotonically decreasing, we can sort it in descending order and keep the sort index (argsort). The ranking according to the optimal linear ranker obtained from sorted position bias can be mapped back to the proper position using the sort index.*

Solution to the MOO problem

In the MOO problem formulated above, the objective functions are not continuous (nor differentiable) since they are functions of ranking. We use a direct search approach to solve the constraints optimization problem. To maintain clarity and be concise, we write the optimization problem in a general form:

$$Min\ f(\mathbf{w})$$
$$s.t.\ g_j(\mathbf{w}) \ge 0, j = 1, 2, \ldots, m$$
$$\mathbf{w} \in [0,1]^{m-1}$$

In the formulation, $f(\mathbf{w}) = \Sigma_{i\le k}\rho(i)\cdot[\gamma(t_{s_i})\cdot r_m(t_{s_i})]$ is the kept objective function from the original m objectives, while $g_j(\mathbf{w}) = \Sigma_{i\le k}\rho(i)\cdot[\gamma(t_{s_i})\cdot r_j(t_{s_i})] - \epsilon_j$ (for $j = 1, \ldots, m-1$) are constraints converted from the rest $m-1$ objectives. In these functions, $\rho(i)$ is estimated bias for position i and s is a ranking decided by $\mathbf{w}$ via $z = \Sigma_{j\le m} w_j\cdot\gamma(t)\cdot r_j(t)$ (where $w_m = \Sigma_{j\le m-1} w_j$). The last constraint $g_m(\mathbf{w}) = 1 - \Sigma_{j\le m-1} w_j = w_m \ge 0$ is added to normalize the weights, which is equivalent to $\Sigma_{j\le m} w_j = 1$ but will reduce the dimensionality of the optimization problem by one.

This constraint optimization can be converted to an unconstrained one through Augmented Lagrangian Multiplier (ALM) function (Bertsekas 2014; Baolin 2005):

$$\phi(\mathbf{w}, \boldsymbol{\mu}, c) = f(\mathbf{w}) + \Sigma_{j\le m}\{[max(0, \mu_j - g_j(\mathbf{w})]^2 - \mu_j^2\}/2c$$

where $\boldsymbol{\mu}$ is Lagrangian multiplier and c is penalty. Powell's method (Baolin 2005) is selected to solve the optimization problem: $Min\ \phi(\mathbf{w}, \boldsymbol{\mu}, c)$ as it is derivative-free and quadratically convergent (it will converge in a finite number of iterations for quadratic objective functions). Based on ALM and Powell's methods the following algorithm is presented to solve the optimization problem of offline simulation for job marketplace ranking.

Algorithm 1: Score based multi-objective ranking

Input: initial point $\mathbf{w}^{(0)} \in [0,1]^{m-1}$,
initial Lagrangian multiplier $\boldsymbol{\mu}^{(1)} \in \mathbb{R}_+^m$,
max number of iterations allowed b

Parameter: upper bound for multipliers μ_{max},
initial penalty $c > 0$,
tolerance $e > 0$,
threshold for penalty update $\alpha \in (0,1)$,
penalty scaling factor $\beta > 1$

Output: optimal solution $\mathbf{w}^*$

define objective and constraint functions:
$f(\mathbf{w}) = \Sigma_{i \le k} \rho(i) \cdot [\gamma(t_{s_i}) \cdot r_m(t_{s_i})]$ and
$g_j(\mathbf{w}) = \Sigma_{i \le k} \rho(i) \cdot [\gamma(t_{s_i}) \cdot r_j(t_{s_i})] - \epsilon_j$
$(j = 1, \ldots, m-1)$ on the basis of the ranking s
decided by score $z = \Sigma_{j \le m} w_j \cdot \gamma(t) \cdot r_j(t)$
and estimated position bias $\rho(i)$.
$g_m(\mathbf{w}) = w_m = 1 - \Sigma_{j \le m-1} w_j$.
Let $n = 1$
while $n \le b$ **do**
  Solve $Min\ \phi(\mathbf{w}, \boldsymbol{\mu}^{(n)}, c)$ using Powell's method (search bounded in [0, 1]) with $\mathbf{w}^{(n-1)}$ as the initial point to get $\mathbf{w}^{(n)}$ (for each iteration the value of ϕ is calculated based on $f(\mathbf{w})$ and $g_j(\mathbf{w})$ for updated $\mathbf{w}$).
  Calculate $h_j^{(n)} = |g_j(\mathbf{w}^{(n)}) - max\{0, c \cdot g_j(\mathbf{w}^{(n)}) - \mu_j^{(n)}\}/c|, j = 1, \ldots, m$.
  if $||\mathbf{h}^{(n)}|| < e$ **then**
    return $\mathbf{w}^{(n)}$ as solution
  else
    Update $\mu_j^{(n+1)} = min\{\mu_{max}, max\{0, \mu_j^{(n)} - c \cdot g_j(\mathbf{w}^{(n)})\}\}, j = 1, \ldots, m$
    if $||\mathbf{h}^{(n)}||/||\mathbf{h}^{(n-1)}|| > \alpha$ **then**
      $c = \beta \cdot c$
    end if
  end if
  $n = n + 1$
end while
return no solution found

Experiments

The optimization method (given by Algorithm 1) is tested on two placements of online job marketplaces owned by our organization. The objective is to identify the best ranker that trades a limited loss of relevance for revenue. The two placements are for different platforms (e.g., web, email, mobile, etc.) and each has its own search engine. Each time the engine retrieves about a thousand relevant jobs and send them as a request for ranking. The first one (placement A) has ∼60 impression positions while the second one (placement B) has ∼25 positions.

The position biases estimated (by the ratio between number of clicks and the sum of relevance score) for the two placements are illustrated in Figure 2. They are calculated for one month of data that are split into two halves to show their change over time.

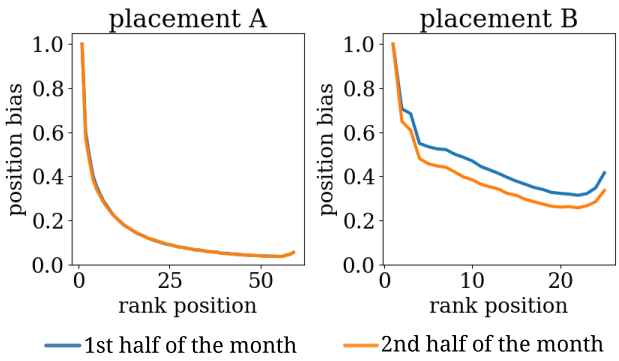

Figure 2: Position bias estimated for placement A and B, which are normalized on the basis of the first position.

As shown in Figure 2 the position bias seems quite stable for placement A, while it drifted a bit for placement B. The drifting suggests that we may need to run simulations frequently for placement B to update the ranker to keep up with marketplace dynamics. The position bias of placement A are mostly monotonically decreasing over rank position. It is not the case for placement B and we will need to sort the position bias prior to simulation and use the sort index to map generated ranking back (see Remark 2).

As a stress test for robustness, Algorithm 1 is applied to the following optimization problem for an offline simulation on placement A.

Problem 1: *maximize the number of clicks for a special type of jobs with total number of clicks >96% of the max possible number of clicks, revenue >75% of the max possible revenue, and promoted job revenue >56% the max possible promoted job revenue.*

It is noted that we can achieve the maximum clicks by ranking via relevance score (i.e., max click ranker), maximum revenue by ranking via bid-relevance product (i.e., max revenue ranker), and maximum promoted job revenue by ranking via the product of bid, relevance, and a binary indicator function (that identifies whether a job is promoted or not). This problem presents a significant challenge since the special jobs account for a very small percentage (<6%) of all jobs. There will be a huge difference in clicks between ranking them to fill all impression positions (∼60) and none of them to these positions (this can be done due to their scarcity). Therefore the optimization problem is not numerically stable (i.e., ill-conditioned) and a small change of the ranker weight will result in a big difference in the objective function value. Algorithm 1 is applied to this problem with the following parameter setting: initial penalty

$c = 50$, penalty scaling factor $\beta = 5$, penalty update threshold $\alpha = 0.5$, and upper bound of multiplier $\mu_{max} = 40$. These settings are suggested as suitable initial parameters for ALM by previous studies (Bertsekas 2014). The best ranker for the problem is obtained in five iterations: $z(t) = 0.506\gamma(t) + 0.164\gamma(t)\cdot bid(t) + 0.132\gamma(t)\cdot bid(t)\cdot I_{t\in P}(t) + 0.198\gamma(t)\cdot bid(t)\cdot I_{t\in Q}(t)$, where $I_{t\in P}(t)$ and $I_{t\in Q}(t)$ are the indicator functions for promoted jobs (P) and the special type jobs (Q). The algorithm has an ending penalty of 6.25×10^3 and the largest multiplier capped at 40. For a comparison a (exterior point) penalty method (Baolin 2005) is applied to the problem, for which we have to increase the penalty to 1×10^6 while scaling down the objective function by 100 to be able to find the optimal ranker of reasonable accuracy (1×10^{-3}).

As an evaluation for the quality of the offline simulation, Algorithm 1 is applied to find a ranker that maximizes three objectives: total click, total revenue, and total revenue of jobs that are promoted for certain business needs. This MOO problem is converted into the following constrained optimization problem:

Problem 2: *maximize revenue with number of clicks >98% of max possible number clicks and promoted job revenue >85% the promoted job revenue of max revenue ranker.*

For placement A, the optimal ranker identified is: $z(t) = 0.751\gamma(t) + 0.135\gamma(t)\cdot bid(t) + 0.114\gamma(t)\cdot bid(t)\cdot I_{t\in P}(t)$, while $z(t) = 0.365\gamma(t) + 0.358\gamma(t)\cdot bid(t) + 0.277\gamma(t)\cdot bid(t)\cdot I_{t\in P}(t)$ is found best for placement B.

The two rankers are tested online for one week to check how well the online click and revenue metrics agree with offline results. In order to facilitate the comparison the metric values are normalized by the corresponding values from a reference ranker. The normalized online and offline metrics values are summarized in Table 1, which shows they agree very well with each other.

Plac.†	Off/On-line	Click	Rev.	PJ. Rev.‡
A	Offline	102.89%	87.03%	79.51%
A	Online	105.72%	88.90%	78.95%
B	Offline	100.95%	97.29%	93.59%
B	Online	99.75%	99.77%	96.89%

Table 1: Normalized click and revenue of offline and online. († Plac. - Placement, ‡ PJ. Rev. - revenue of promoted jobs)

In particular, close approximation is observed even for placement B of drifted position bias. It indicates the rankers identified in simulation by the presented optimization approach are able to accurately influence ranking in favor of desired business outcomes.

Conclusions

In this work, a robust multiple objective ranking approach is presented for identifying the best ranking function that can arrive at the desired balance of multiple business objectives. The approach is based on an unbiased estimate of position bias and a direct search algorithm that integrates Augmented Lagrangian Multiplier method and Powell's method. The approach demonstrates robustness in solving a numerically ill-conditioned optimization problem configured per real business need. In addition, an online test of rankers obtained from offline simulation shows a close agreement between the online and offline metrics values even when there is a slight drift of position bias. The positive online evaluation results indicate the identified rankers can accurately influence ranking in favor of desired business outcomes. The ability of the marketplace optimization approach in handling ranking for objectives that are subjected to fast and continuous refining is critical to online marketplaces competing effectively in the crowded and dynamic environment.

References

Bai, G.; Xie, Z.; and Wang, L. 2018. Practical Constrained Optimization of Auction Mechanisms in E-Commerce Sponsored Search Advertising. arXiv:1807.11790.

Baolin, C. 2005. *Optimization theory and algorithms (2nd Chinese Edition)*. Tsinghua University Press.

Bertsekas, D. P. 2014. *Constrained optimization and Lagrange multiplier methods*. Academic press.

Bi, K.; Teo, C. H.; Dattatreya, Y.; Mohan, V.; and Croft, W. B. 2020. Leverage Implicit Feedback for Context-aware Product Search. arXiv:1909.02065.

Dai, X.; Hou, J.; Liu, Q.; Xi, Y.; Tang, R.; Zhang, W.; He, X.; Wang, J.; and Yu, Y. 2020. U-rank: Utility-oriented learning to rank with implicit feedback. In *Proceedings of the 29th ACM international conference on information & knowledge management*, 2373–2380.

Emmerich, M. T.; and Deutz, A. H. 2018. A tutorial on multiobjective optimization: fundamentals and evolutionary methods. *Natural computing*, 17: 585–609.

Goswami, A.; Zhai, C.; and Mohapatra, P. 2018. Towards Optimization of e-Commerce Search and Discovery. *ecom@ sigir*, 2.

Hua, Y.; Liu, Q.; Hao, K.; and Jin, Y. 2021. A survey of evolutionary algorithms for multi-objective optimization problems with irregular Pareto fronts. *IEEE/CAA Journal of Automatica Sinica*, 8(2): 303–318.

Li, Q.; Wang, L.; Xia, L.; Zheng, W.; and Zhou, Y. 2023. A practical multi-objective auction design and optimization framework for sponsored search. *Operations Research Letters*, 51(6): 541–547.

Sharifi, M. R.; Akbarifard, S.; Qaderi, K.; and Madadi, M. R. 2021. A new optimization algorithm to solve multi-objective problems. *Scientific Reports*, 11(1): 20326.

Tang, W.; and Tang, F. 2023. The Poisson binomial distribution—Old & new. *Statistical Science*, 38(1): 108–119.

Wang, Y.; Wei, B.; Yan, J.; Chen, Z.; and Du, Q. 2012. Multi-objective optimization for sponsored search. In *Proceedings of the Sixth International Workshop on Data Mining for Online Advertising and Internet Economy*, 1–9.

Zaizi, F. E.; Qassimi, S.; and Rakrak, S. 2023. Multi-objective optimization with recommender systems: A systematic review. *Information Systems*, 117: 102233.