Optimizing for Member Value in an Edge Building Marketplace

Optimizing for Member Value in an Edge Building Marketplace

Ayan Acharya
ayacharya@linkedin.com
LinkedIn Inc.
Sunnyvale, California, USA

Siyuan Gao
sigao@linkedin.com
LinkedIn Inc.
Sunnyvale, California, USA

Ankan Saha
asaha@linkedin.com
LinkedIn Inc.
Sunnyvale, California, USA

Borja Ocejo
bocejo@linkedin.com
LinkedIn Inc.
Sunnyvale, California, USA

Kinjal Basu
kinjal@alumni.stanford.edu
Aliveo AI
Austin, Texas, USA

Keerthi Selvaraj
keselvaraj@linkedin.com
LinkedIn Inc.
Sunnyvale, California, USA

Rahul Mazumder
rmazumder@linkedin.com
LinkedIn Inc. (MIT)
Sunnyvale (Cambridge), USA

Aman Gupta
amagupta@linkedin.com
LinkedIn Inc.
Sunnyvale, California, USA

Parag Agrawal
pragrawal@linkedin.com
LinkedIn Inc.
Sunnyvale, California, USA

ABSTRACT

Social networks are prosperous marketplaces where creators and consumers congregate to share and consume various content. In general, products that rank content for distribution (such as news-feeds, stories, and notifications) and are related to edge recommendations (such as connect to members, follow celebrities or groups or hashtags) optimize the experience of active users. Typically, such users generate ample interaction data amenable to accurate model training and prediction. In contrast, we prioritize enhancing the experience of inactive members (IMs) who do not have a rich connection network. We formulate strategies for recommending superior edges to help members grow their connection network. Adapting the recommendations provides enormous value to the IMs and can significantly influence their future behaviour and engagement with the ecosystem. To that end, we propose a general and scalable multi-objective optimization (MOO) framework to provide more value to IMs as invitation recipients on LinkedIn, a professional network with over $900M$ members. To deal with the enormous scale, we formulate the problem as a massive constrained linear optimization involving billions of variables and millions of constraints and efficiently solve it using accelerated gradient descent, *making this the largest deployment of LP-based recommender systems worldwide.* Furthermore, the proposed MOO paradigm can solve the general problem of matching different types of entities in an m-sided marketplace. Finally, we discuss the challenges and benefits of implementing and ramping our method in production at scale at LinkedIn and report our findings about the core business metrics related to users' engagement and network health.

CIKM '23, October 21–25, 2023, Birmingham, United Kingdom

ACM ISBN 979-8-4007-0124-5/23/10…$15.00
https://doi.org/10.1145/3583780.3615000

CCS CONCEPTS

• **Mathematics of computing → Linear programming**; **Convex optimization**; • **Information systems → Social networking sites**; **Social recommendation**; • **Computing methodologies** → *Neural networks.*

KEYWORDS

second-pass-ranker, multi-objective optimization, people recommendation, linear programming

ACM Reference Format:
Ayan Acharya, Siyuan Gao, Ankan Saha, Borja Ocejo, Kinjal Basu, Keerthi Selvaraj, Rahul Mazumder, Aman Gupta, and Parag Agrawal. 2023. Optimizing for Member Value in an Edge Building Marketplace. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management (CIKM '23), October 21–25, 2023, Birmingham, United Kingdom.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3583780.3615000

1 INTRODUCTION

Social networks significantly influence how people interact over the internet with their peers. The ever-growing popularity of such platforms is a testament to their ability to act as a fundamental medium for public discourse. Other popular communication methods include messaging and email. The mode of communication is the central distinguishing factor between social networks and messaging/email apps. The primary communication method in social networks is broadcasting information, typically to a user network or groups and communities [33]. This paper focuses on the broadcasting use case in social networks. Our setting is an ecosystem with edges forming between members where the content generated by the *creators* reaches the *consumers* in their network. As more creators manufacture content of diverse nature, consumers have more variety and liquidity to choose from. Consequently, they find more value in the social network and visit more frequently.

Effective network formation is paramount for building proper connections and nurturing richer conversations between members in such an ecosystem. Justifiably, a rich body of work, which we address as *edge building recommendation systems*, delves into how to assist consumers in growing their connection network efficiently

[6, 43]. Most social network platforms leverage such research to enhance the engagement of the members. These systems generally deal with multiple objectives, such as network growth, construction of a quality network, and retention improvement. These objectives are often at odds and accompanied by multiple practical constraints that must be accounted for methodically.

Unfortunately, a disproportionately large number of members on such platforms are not proactive about forming the right connections. The abundance of interactions from the more active users usually overwhelms the logged data; therefore, models trained based on such biased data endorse forming connections between active members, reinforcing the *rich-get-richer* cycle [14, 31, 34, 41]. Consequently, the *inactive members* (IMs) receive poor content liquidity and quickly lose interest in the platform. We hypothesize that many such IMs may reengage with the platforms if they receive relevant invitation requests [46]. Once these members form a healthy network, a better and regular flow of content follows, keeping them engaged and encouraging them to visit more frequently.

This paper presents techniques for providing value to IMs by delivering relevant invitation requests. We propose a novel linear programming (LP) [9, 10] based solution to model the LinkedIn social network with over 900 million users. The resulting LP formulation is challenging to optimize because of scaling reasons. We customize DuaLip [8, 39] to solve this optimization problem with billions of variables and provide a complete recipe to modify the re-ranker to promote the IMs. To our knowledge, this is the largest deployment of LP-based recommendation systems in the industry. Below, we summarize our principal contributions:

- We present a general framework where we represent the edge-building recommendation problem as a two-sided marketplace, considering the value of the IMs as recipients of invitations.
- We demonstrate how to formulate the constrained MOO problem as an LP. We thoroughly describe how to adapt a general-purpose toolkit such as DuaLip and customize the process of data generation, hyper-parameter tuning, data compression, offline validation and online experiments to provide targeted promotions to IMs.
- *This is the first-ever paper that navigates through all the complexities, scaling challenges and uncertainties related to deploying an LP-based re-ranking solution in a production system serving* 850 *million LinkedIn users.* We exploit distributed inference and fast projection operations to handle such a gigantic scale. We also report how the deployed system influences core business metrics related to user engagement and the long-term health of the ecosystem.
- The optimization paradigm solves the general problem of matching entities of two different types, which can find applications in many recommender system applications in an m-sided marketplace.

2 RELATED WORK

Several scientific and industrial disciplines benefit from large-scale LP [9, 10] formulations of underlying problems. Such problems also abound in the internet industry in different forms ranging from optimizing the volume of emails sent to users [22, 23] and matching entities to consumers in an *m*-sided marketplace [7, 50] to balancing multiple business metrics in recommender systems [2–4]. Solving these large-scale problems using generic or commercial solvers [24] is impossible. The people recommendation problem discussed in Section 4 involves billions of variables beyond the capacity of generic solvers that use simplex or interior point methods [21]. Unless the problem structure is carefully exploited, these methods become prohibitively expensive for large-scale applications [11, 18].

DuaLip [8, 39] is an open-source LP library that allows users to solve LPs at the scale that the modern internet industry demands. At a high level, DuaLip makes the underlying linear objective strongly convex by introducing a squared regularization term. Such modification guarantees a Lipschitz continuous gradient, making the optimization amenable to gradient descent methods. The primal solution involves vertex-oriented projection on a polytope and the dual ascension uses Nesterov's accelerated gradient descent (AGD) [45] and Limited-memory BFGS-B (LBFGS-B) [12].

Logged feedback data in recommender systems are typically biased for positional representation, specific promotions, homophily, popularity, contemporaneity etc. [28]. Such biases often encourage *rich-get-richer* behaviour [1, 14, 19, 27, 28, 31, 34, 41]. The corresponding bias in people recommendation systems arises because of the lack of representation of the IMs in the logged data. As a result, the interactions of the active users overwhelm the data used to train the ranking models; the IMs' relevance is systematically degraded over time, encouraging connections to be formed only among active users (a sizeable fraction of the IMs on LinkedIn have less than 10% of connections compared to an average active user). The long-term effect of such homophily is disastrous [19]. The network loses its popularity and results in significant user churn.

We adopt DuaLip to address the rich-get-richer bias. Most works on edge-building recommendation systems maximize the value for content consumers[1]. Our work is distinctively different as we consider promoting IMs to ensure equitable engagement from users of varying activity levels. Moreover, DuaLip requires non-trivial integration and customization to cater to business requirements. Though the ideas of contextual multi-arm bandit [5, 15, 44], diversified recommendations [20, 42, 48] and doubly robust policy estimation [29, 30, 37] seem relevant for solving such problems, they often fail in real-world, large-scale recommender systems due to specific practical and business-related challenges, such as hyper-parameter tuning, score calibration, data compression, pruning, and offline validation. Notably, ours is the first-ever approach in the literature that provides a complete recipe for using an LP-based solution to modify the re-ranking formulation for a recommender system. A thorough literature search reveals that nothing implemented so far in the industry or any commercial solver comes remotely close to the scale we deal with.

3 PEOPLE YOU MAY KNOW (PYMK)

The LinkedIn economic graph is a digital representation of the global economy with 1B nodes and 200B edges (sparsity 0.00002%). The PYMK connection graph is a subset of this graph that consists of 900M members. PYMK is generally designed as a bidirectional edge-building recommendation system where users are recommended to other members they can connect to. Let i denote the user being shown the edge recommendations (aka the *source member*),

[1]The only exception is the work by Lo et al. [33] who focus on augmenting creator utilities to the recommendation objective.

and j represent the specific entity with whom i is being recommended to form a connection (aka the *destination member*). When a connection is made, the users can access content from each other through distributional channels like feeds and notifications. Such a recommendation framework has a few primary considerations:

- the probability of an invitation being sent from the viewer (source) i to the recipient (destination) j, $\texttt{pInvite}_{i,j}$
- the probability of accepting the invitation from the viewer i by the recipient j (given that they receive an invitation), $\texttt{pAccept}_{i,j}$
- the value of the edge, if formed, $\texttt{vEdge}_{i,j}$.

At LinkedIn, the final PYMK relevance score has historically been framed as a composite ranker consisting of a combination of the scores mentioned above:

$$\texttt{Score}_{i,j} = \texttt{pInvite}_{i,j} * [1 + \texttt{pAccept}_{i,j}[\alpha + \beta * \texttt{vEdge}_{i,j}]]. \quad (1)$$

This score is a solution to a Lagrangian dual problem of a constrained optimization [33] where α and β are chosen from running multiple online experiments.

At LinkedIn, the probability scores and edge-value are modelled using supervised learning frameworks [32]. We train separate models on logged interaction data to derive each score. These models are referred to as *first-pass-rankers* (FPRs) as they produce the first set of relevance scores for a given source-destination pair. In particular, the `pInvite` FPR optimizes the source-side experience to encourage users to send out more invitations to members they may want to connect with. On the other hand, the `pAccept` utility addresses the receiver-side behaviour so that the receivers are encouraged to accept more invitations from sources they would like to connect to [32]. `vEdge` represents a utility that prioritizes connections that may lead to increased destination session activity [36]. The FPRs consume many graph-based features (such as personalized-page-rank scores, triadic closing scores, graph-based user embeddings etc.), member attributes and actions (such as job description, educational qualifications, geo-locations, interactions with feed etc.) to generate relevance scores for each source-destination pair.

The current composite ranking formulation has distinct disadvantages. Initially, the edge-building recommendation problem was designed to optimize for the source-side objectives. Such a strategy exacerbates the *rich-get-richer problem*, with most sources seeing the popular candidates as recommendations. Furthermore, the existing system fails to incorporate evolving business constraints to adapt these recommendations. The bias in the training set aggravates the rich-get-richer problem, where active members receive more invitations, and IMs lose out on opportunities to connect. This phenomenon results in a cycle of degrading the IMs' experience.

The PYMK team tried two different approaches earlier to promote the IMs. In the first instance, we randomly promoted IMs in a user session. Unfortunately, such random shuffling of the slate does not produce consistent results, as the specific characteristics of the source-destination pairs are not considered. We also formulated a retention utility model to prevent IM churn and engage them better on the platform. This retention utility model served as an FPR and produced another utility score for each source-destination pair. This score was added as a weighted term to Eq. 1. Sadly, we did not see any improvement in IM retention.

The proposed LP formulation is our *first successful attempt* to promote the IMs. Unlike previous efforts, the constraints on the expected number of invitations received directly and coherently impact the promotion of the IMs. We formulate this problem as a two-sided marketplace where we simultaneously optimize the objectives of both the source and destination members and provide a more equitable distribution of exposure to the IMs.

Finally, the industry-wide practice [3, 4] for scoring relevance according to Eq. 1 and with hand-tuned parameters α and β often leads to a sub-optimal ranking and loss in developers' productivity. Considering $R > 1$ different utilities and T different settings of each weight, the grid search for optimal configuration runs over $(R-1)T$ values. Since these configurations must be evaluated online, they block traffic from online serving, leading to expensive human labour and computation costs. Hyper-parameter optimization techniques [17] can be used for solving such problems, but they often require a non-trivial implementation to work in conjunction with legacy systems. The proposed LP formulation can also be adapted to estimate these parameters, the detailed discussion of which is beyond the scope of this paper.

4 PROBLEM FORMULATION

At a high level, the proposed framework maximizes the expected combined relevance score over all source-destination pairs while ensuring the following:

- The cumulative impression associated with the recommendations for the IMs by each source member is less than a given value.
- The expected number of invites sent to each IM does not drop below a certain threshold.

As shorthand, we use p_{ij} to refer to $\texttt{pInvite}_{i,j}$, q_{ij} to refer to $\texttt{pAccept}_{i,j}$ and r_{ij} to refer to $\texttt{vEdge}_{i,j}$. Note that $p_{ij}q_{ij}$ refers to the probability of the formation of an edge between the members i and j. We formulate the task of providing equitable exposure to upcoming IMs, via the following LP:

$$\begin{aligned} \max \sum_{i \in \mathcal{S}} \sum_{j \in \mathcal{T}_i} & x_{ij} p_{ij} (1 + q_{ij}(\alpha + \beta r_{ij})) \\ \text{s.t.} \sum_{j \in \mathcal{T}_i} & x_{ij} \leq \delta_i \; \forall i \in \mathcal{S}, \; 0 \leq x_{ij} \leq 1, \\ \sum_{i \in \mathcal{S}} & x_{ij} p_{ij} \geq \zeta, \; \forall j \in \mathcal{T} \end{aligned} \quad (2)$$

where, $\{x_{ij}\}$'s are the decision variables. x_{ij} represents the probability of the j^{th} destination member appearing in the recommendation list of the i^{th} source member. $\mathcal{S}$ is the set of all source members, and $\mathcal{T}_i$ the set of all destination members who appear in the recommendation of the i^{th} source member. In our implementation, $\mathcal{T} = \cup_{i \in \mathcal{S}} \mathcal{T}_i$ consists of only a predefined set of IMs, so we focus on optimizing the interactions among all possible source members (that include both FMs and IMs) and IM destination members only.

Using formulation (2), one can populate the online recommendations of the source members with candidates of higher relevance scores from the set $\mathcal{T}$ to facilitate a more equitable distribution of connections. The constraint $\sum_{j \in \mathcal{T}} x_{ij} \leq \delta_i$ implies that no member can monopolize the system by sending too many invites, so the rich-gets-richer problem is scrupulously mitigated. The second constraint ensures that each IM receives a minimum number of invitations. As explained below, the dual variables corresponding to the second set of constraints contribute to personalized weights and cater to customized recommendations. DuaLip optimizes the following problem with a strongly convex objective:

$$\begin{aligned}&\min \sum_{ij} \left(\frac{\gamma}{2}x_{ij}^2 - x_{ij}p_{ij}(1+q_{ij}(\alpha+\beta r_{ij}))\right)\\ &\text{s.t.} \sum_{j\in\mathcal{T}_i} x_{ij} \leq \delta_i \ \forall i \in \mathcal{S},\ 0 \leq x_{ij} \leq 1,\\ &\sum_{i\in\mathcal{S}} x_{ij}p_{ij} \geq \zeta,\ \forall j \in \mathcal{T}.\end{aligned} \tag{3}$$

Note that the only departure from (2) is the addition of the squared term $\sum_{ij}(\gamma/2)x_{ij}^2$ in the objective, which can be beneficial from a computational standpoint [8, 38, 39]. The box-cut constraint is explicitly modelled in the primal projection step of DuaLip, so there is no need to learn the duals corresponding to that constraint. Here δ_i corresponds to the total measure of the impression the i^{th} source member makes. We use a parametric formulation to derive δ_i as:

$$\delta_i = f(\sum_{j\in\mathcal{T}_i} y_{ij});\ f(y) = y\mathbb{I}(\{y \leq \epsilon\}) + (\epsilon + (y-\epsilon)/\rho)\mathbb{I}(\{y > \epsilon\})$$

where $\mathbb{I}(.)$ refers to the indicator function and ϵ and ρ are hyper-parameters of this problem setting.

The above parametric formulation penalizes sending out too many invitations from source members. y_{ij} is an indicator variable that represents if the j^{th} target number is recommended to the i^{th} source member. When the aggregated value of this number, given by $\sum_{j\in\mathcal{T}_i} y_{ij}$, exceeds the threshold ϵ, we discount those values by a scale of ρ, which prevents source members with many potential connections to IMs from sending out more invitations. In Section 6.2, we simplify this bound further and provide a detailed description regarding how it controls the dynamics of the optimization.

The Lagrangian of formulation (3) can be written as:

$$\begin{aligned}\mathcal{L}(\boldsymbol{\lambda}, \mathbf{x}) = \sum_{i\in\mathcal{S}}\sum_{j\in\mathcal{T}_i} &\Big(-x_{ij}p_{ij}(1+q_{ij}(\alpha+\beta r_{ij})) + \frac{\gamma}{2}x_{ij}^2\\ &+\lambda_j(\zeta - x_{ij}p_{ij})\Big).\end{aligned} \tag{4}$$

From optimality conditions, the optimal primal (x_{ij}^*s) and dual (λ_j^*s) variables will satisfy:

$$x_{ij}^* = \pi\left(\frac{1}{\gamma}p_{ij}(1+\lambda_j^* + q_{ij}(\alpha+\beta r_{ij}))\right), \tag{5}$$

where $\pi(.)$ represents the (Box-cut) projection operator [8, 38] that produces a score in the $[0, 1]$ interval. Comparing with the score in (1), it can be seen that we merely boost the base score by an additive factor of $p_{ij}\lambda_j^*$ – the scores of all other (source, destination) pairs remain the same as Eq. 1. The higher the value of λ_j^*, the higher is the relevance of the corresponding IM for a given source member. For online implementation, one does not need to worry about scaling and projection as long as the ordering of the scores (and not the absolute values) is relevant. Note that the λ_j's learned from the LP provide the desired personalization in this altered formulation.

Note that promoting all the IMs in the ecosystem is computationally infeasible and liquidity-wise improbable. Many members are not responsive and hence cannot be resurrected easily. We consider different gradations of IMs, based on user behaviour (like frequency of visits) and model estimates (like the propensity of these members to engage with the platform) if they are provided incentives like more invitations [49] and use it to finally populate the set $\mathcal{T}$.

5 BUSINESS IMPLICATIONS

The objective of the LP-based formulation is to consume the relevance scores computed by the FPRs and compose a final score that considers the specific constraints the business demands and balances the utilities offered by the FPRs. This flexibility and decoupling are desirable in modern-day recommender systems where at a given point in time, the business may require specific cohorts of users to be handled differently, and the FPRs undergo frequent updates. For example, the IMs getting prioritized in favour of the ecosystem's overall health is a business decision driven by many product strategies. Similarly, the input attributes used in the FPR models change based on the new product features that get continually added. Model architectures and optimization algorithms also change frequently based on the latest developments in academia and industry. Therefore, the re-ranking layer and the training process of the FPRs is often kept decoupled in the Linkedin ecosystem.

From the product management perspective, maintaining parity with the existing SPR formulation (Eq. 1) while promoting the IMs in a mathematically consistent way is challenging. The legacy PYMK system admonishes any disruptive changes, so one has to comply with the business constraints and infrastructural challenges. The proposed solution does not add significant computing or memory burden to the backend system that serves the recommendations in real-time. As we explain more in Section 7.2, we push only a single float value to the online feature store for each destination member, which needs to be fetched during online scoring in real-time and to be used in Eq. 5. The simplicity and elegance of the re-ranking score derived from a rigorous mathematical formulation is a desirable feature for modern-day recommendation systems.

The proposed optimization framework is generic enough to find applications in constructing re-rankers, typical in most modern-day recommender systems. Such re-rankers can be cast as large-scale problems of matching entities of different types, which we can formulate as LP. One only needs to represent the business metrics in primal variables and change the constraint according to the problem definition. For example, to ensure a certain amount of notifications being exchanged between users, we can add a constraint where the total number of notifications needs to meet a threshold.

6 SYSTEM DESCRIPTION

The system architecture described in this section has two components. First, the offline flow solves the optimization problem at an enormous scale; the online flow modifies the relevance scores for the source-destination pair according to the duals derived by the offline flow and serves the recommendation in real-time. We construct the offline system in two different stages. In the first stage (Approach 1 detailed in Section 6.1), we attempt to run DuaLip in a subset of the tracking data. Then, in the second stage (Approach 2 detailed in Section 6.2), we adapt and customize DuaLip to work with the entirety of the tracking data.

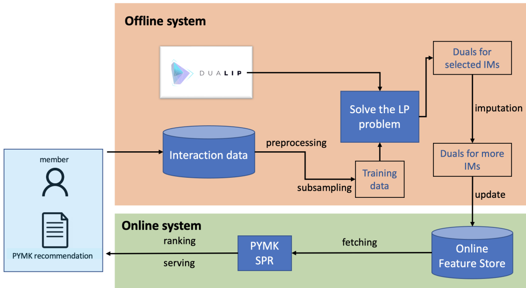

Figure 1: System Diagram for Approach 1

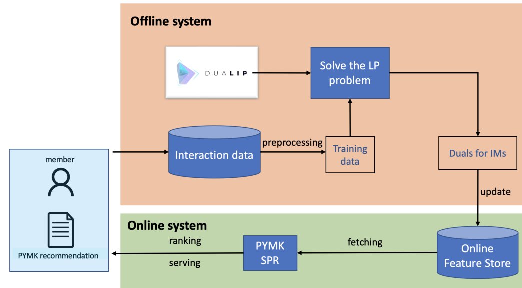

Figure 2: System Diagram for Approach 2

6.1 Offline System – Approach 1: Clustering and Dual Extrapolation

6.1.1 Managing the scale of the PYMK problem. Even though DuaLip is designed to solve problems of enormous scale, the sheer number of primal and dual variables associated with the proposed formulation poses a significant challenge to its adoption. The scale of the problem grows linearly with $|\mathcal{S}|$ and $|\mathcal{T}|$. The PYMK connection graph is fairly sparse; hence the scale of the problem is never dictated by the product of $|\mathcal{S}|$ and $|\mathcal{T}|$. Rather, it is governed by the average size of the recommendations times $|\mathcal{S}|$. Regardless, this number is too large for DuaLip to handle.

Unfortunately, reducing the scale of the problem is not straightforward. If we arbitrarily sample members from the set $\mathcal{T}$, the characteristics of the original data get severely distorted. Another idea is to reduce the size of the set $\mathcal{T}$ by selecting IMs with a higher likelihood of responding to invitations. The PYMK system has few predictive models that can prioritize the set of IMs who may accept more invitations and engage in meaningful conversations and activities on the platform. However, such models have limitations, as they are often trained on logged data with lower IM representation; hence, they fail to apprehend the nuances of how the IMs behave.

6.1.2 Clustering of IMs. We made several attempts in the past to cluster the IMs. However, the large, dynamic, ever-growing user base makes such clustering efforts challenging to scale, maintain and retrain. Considering the successful applications of graph-neural networks at such a large scale [25], we tried building member embeddings from the connection graph and profile attributes. Our key challenge is the problem associated with graph isomorphism, which we formally describe in Walker et al. [47]. The IMs generally have sparse connections and live in isolated parts of the graph. More importantly, due to the limited number of connections they possess, the local neighbourhood becomes similar. As a result, two dissimilar IMs with structurally similar local neighbourhoods can get arbitrarily close to each other in the embedding space. Therefore, any graph-affinity-based clustering idea is challenging to implement for these IMs. We have successfully built representations for the more active members though, who usually have detailed information in their profiles and are connected to a denser network.

If the IMs could be partitioned into smaller coherent subsets using some measures, one could solve the optimization problem corresponding to each subset and then summarize the duals obtained using some form of aggregation, such as max-pool, mean-pool, etc. To that end. we hypothesize that the IMs can be categorized based on the pairwise scores $\{p_{ij}\}$ corresponding to different source members. Naturally, it might be tempting to represent the j^{th} IM with a vector of size $|\mathcal{S}|$ as $\mathbf{p}_j = (p_{ij})_i$. However, there are two fundamental challenges with such representation. Firstly, $|\mathcal{S}|$ can be of the order of a few tens to hundreds of millions, making such representation extremely high-dimensional and sparse. More importantly, for any two IMs j_1 and j_2, selected at random from the graph, $\langle \mathbf{p}_{j_1}, \mathbf{p}_{j_2} \rangle$ is almost always zero. The IMs, on average, are connected to only a few members. Hence, the common connection of any two randomly selected IMs is almost always a null set.

To alleviate such issues, we compare two IMs based on the distribution of the associated $\{p_{ij}\}$ scores. We partition the $\{p_{ij}\}$ scores of the j^{th} IM into ten bins uniformly distributed between 0 and 100. As a result, the j^{th} IM gets represented by a 10-dimensional vector $\mathbf{v}_j$. Each component of such a vector represents the fraction of source members for which the corresponding $\{p_{ij}\}$ scores belong to that bucket. In other words, the continuous distribution of the $\{p_{ij}\}$ scores is approximated using a discrete distribution, so comparing the distributions of two IMs is easy. Arguably, such representation has its limitation, as the position of an IM in the graph is completely ignored. The strong empirical evidence (presented in Section 7.2.3) suggests that such a representation is a reasonable baseline that one can improve upon.

Further, to reduce the problem-size, we sample the set of IMs based on the values of $\{\mathbf{v}_j\}$. We first perform a k-means clustering with $\{\mathbf{v}_j\}$ used as features. Consider that we need to sample M IMs from the pool of $|\mathcal{T}|$ IMs ($M << |\mathcal{T}|$). The clustering provides a rather consistent framework to sample such IMs systematically. Suppose the k-means algorithm is run with K clusters and J_k IMs are assigned to the k^{th} cluster. In a stratified sampling implementation, we then select approximately $m_k = J_k M / |\mathcal{T}|$ IMs from the k^{th} cluster. To make the sampling relevant, we choose IMs closer to the centroid of the clusters with a higher probability. We use this set of IMs $\{m_k\}_{k=1}^{K}$ in the optimization framework. An appropriate choice of M and K might produce duals that do not deviate much from a solution to the original problem if solved at full scale. In our implementation, M is set to 300K, and K is set to 100. The number of FMs associated with these IMs is approximately 2.5M.

6.1.3 Imputation of duals. Though the process of clustering and subsequent sampling intuitively make sense, one still needs to impute the dual values corresponding to the IMs not selected in the sampling process. We speculate that IMs with similar representation in the 10-dimensional feature space must have similar dual values. To that end, we resort to the nearest neighbour algorithm. For the j^{th} IM, the corresponding λ_j can be calculated as

$\lambda_j = 1/|\mathcal{N}_j| \sum_{j' \in N_j} \lambda_{j'}$ where $\mathcal{N}_j$ refers to the set of IMs selected in the sampling process. We consider only the top−5000 neighbours for imputing the duals. The proximity is derived according to the 10-dimensional feature representation described above.

6.2 Offline System – Approach 2: Solving the Entire Problem

6.2.1 Scaling DuaLip. Though the approach presented in Section 6.1 seems appealing, it cannot produce enough positive-valued duals to move core business metrics. As we explain the results of our first set of ramps in Section 7, a statistically significant movement in these metrics requires a good fraction of the IMs to have positive-valued duals. Desperate to solve the problem in its original scale, we investigate and amend the following bottlenecks of DuaLip:

- The critical computation bottleneck of the optimization technique employed in DuaLip arises from the cost of primal projection. The current formulation requires a box-cut projection ($\sum_j x_{ij} \leq \delta_i$), which we implement using a variant of the polytope projection algorithm [39], where each executor in a distributed setting computes the so-called corral-set [13] to project onto a different data block, resulting in as much as eight times speedup compared to generic projection algorithms [16].
- The dual ascension step of DuaLip is carried out in the driver after aggregating the gradients from all different executors. Adopting a two-step gradient aggregation scheme makes this stage highly efficient for hundreds of millions of constraints. In the first step, we partition the dual vector into several sub-vectors of similar sizes, where the number of partitions is a user-provided argument. We perform aggregation of gradient components corresponding to these sub-vectors in different executors. Eventually, we construct the gradient of the entire dual vector in the driver by aggregating information from such executors. This results in 2-3 times performance improvement per iteration of primal-dual optimization.
- For analyzing the convergence and the consistency of the step size with the gradient, DuaLip saves the status of the optimizer after every iteration. These logs are essential for developers as they show the violations of the constraints, the value of the objective, the regularization term, slack, and the number of active constraints. The collection and documentation of these logs require fast information aggregation from all the executors. We employ an efficient accumulation of such information from multiple workers, making this computation negligible compared to the primal projection step.

With these improvements, DuaLip solves the original problem at full scale without the clustering and extrapolation components of Approach 1. Since we can generate positive-valued duals corresponding to a healthy fraction of IMs, we do not have to rely on ad-hoc strategies to accumulate duals over multiple days. This offline workflow runs within six hours (see Fig. 2).

6.2.2 Effect of Box-cut Constraints on Dual Distribution. For this implementation, we choose a single value for δ_i for the box-cut constraint instead of the parametric form given in Eq. 4. The choice of δ_i has a significant impact on the optimization. A higher value of δ_i implies that all the source members would send more invitations to the IMs. As a result, the constraint that ensures that each IM should receive a minimum number of invitations in expectation gets easily satisfied. The duals corresponding to such values remain low, and the corresponding IMs have less chance of being promoted.

On the other hand, a lower value of the duals implies that each source member sends out more occasional invitations to the IMs. The corresponding constraints become pretty challenging to be satisfied, and hence the duals increase in values. Higher values of the duals ensure that the IMs are promoted more in the recommendation list. Essentially, the choices of the values of δ_i's dictate two extreme settings. In one case, when the δ_i's are sufficiently small, the IMs get promoted so much that the experience of the source members is completely compromised. When the δ_i's are appropriately large, the IMs do not receive any promotion, and the solution turns out to be no different from the baseline recommendation strategy presented in Eq. 1. In Section 7.2.5, we demonstrate how different choices of δ_i's influence the dual distribution.

6.3 Implementation Details

We describe the implementation required for Approach 1 and how a large part of it is simplified for Approach 2. Since we solve the problem for only a small subset of IMs selected via the clustering and sampling process in Approach 1, we need to extrapolate the solution to other IMs not selected in the sampling process. The offline part of the workflow (see Fig. 1) parses the interaction data from the platform, filters information that is pursuant to the optimization formulation, samples the IMs and prepares the data required by DuaLip, ingests the dual values that the solver derives, and extrapolates these values to all the IMs who are not selected in the sampling process. We use $\epsilon = 2$ and $\rho = 50$ in Eq. 4 as these values provide a more equitable distribution of the duals we desire. We set $\zeta = 1$ and run the workflow daily to avail new interaction data and keep timely values of λ_j's for the most relevant IMs.

Note that, at the imputation stage, we join two large tables – one containing the attributes of the IMs selected in the sampling process (of size $300K \times 10$) and the other having the attributes of the IMs not selected in the sampling process. First, the attributes of the smaller table are compared with those in the larger table using squared Euclidean distance. The source members are then grouped by the destination members and ordered by the similarity score, after which the top 5000 most similar IMs are retained from the source table. We then perform a simple averaging of the duals associated with the IMs from the source table to derive the duals for the IMs in the destination table. Note that the above run times are relevant for the offline system from Approach 1.

Since the workflow runs daily, we parse the interaction data over a range of two weeks before the execution date to keep the memory and computation footprint manageable. We can only derive the dual variables corresponding to the IMs that appear in such two-week-long data. Since the IMs visit a social media platform only once every few weeks, a considerable fraction of them would be missing in a two-week-long window. We adopt a simple strategy to compute the duals for most of the IMs. If an IM receives a positive dual, we persist in the value and do not change it.

Since our final objective is to promote the IMs, one may be tempted to retain the greatest of all the duals discovered for any given IM derived on any day. We run some experiments with such a strategy as well. In section 7.2.2, we explain how such an accumulation of duals results in many of the IMs, much more than what we anticipate, getting promoted higher in the recommendation list, which creates a negative experience for the source members. Overwhelming the source members with the recommendations of too many IMs deters them from sending out invitations.

6.4 Online System

The online workflow collects the data from the table consisting of all the duals and pushes them to an online venice store which serves the required duals during a ranking stage. Note that such ranking is performed on a small subset of members that many candidate generation mechanisms (offline and online) find relevant. Whenever a source member visits the website, a PYMK end-point is invoked, which subsequently ranks the candidates based on the relevance scores computed according to Eq. 5. The online implementation ensures that all relevant computations, including fetching the duals from the online store and (re)scoring the IM candidates, are done within a latency budget of a few tens of milliseconds.

As we further explain in Section 7, performing an A/B test on a connection graph of a social network poses its unique set of challenges. Unlike in other applications where A/B testing is employed, the *Stable Unit Treatment Value Assumption* (SUTVA) [26, 40] is easily violated in a network model, as information leaks through the shared connections between the treatment and control group. We implement an attention-balancing mechanism in the online workflow of the PYMK system to account for such violations [33]. At a high level, for every single destination candidate, we maintain two different scores – one that is derived according to the control model and the other according to the model that boosts IMs. Finally, we merge such scores and alter the rank orders according to Algorithm 1 as described in Nandy et al. [35]. Such correction ensures that our metrics, as observed from the perspective of the destination members, are relatively free from unobserved confounders.

7 EXPERIMENTS

To assess the quality of the solution, we conduct offline analysis and online A/B tests. The offline study provides an initial estimation of the quality of the dual variables. The online A/B tests are used to fine-tune the hyper-parameters further and evaluate the actual impact on the users. Note that the *control* traffic in our online A/B tests is the re-ranking strategy dictated by Eq. 1. The *treatment* traffic is exposed to the re-ranking formulation given in Eq. 5. We use $\alpha = 1.0$, $\beta = 2.8$, and $\gamma = 0.001$ in all of our experiments.

7.1 Offline Analysis

7.1.1 Analysis of the Duals and Violations. We need to assess the quality of the LP solution before using them in the online production system. The convergence of the objective value of LP conveys partial information about the quality. The dual values may have a skewed distribution even when LP has the desired convergence. Such uneven distribution leads to unsatisfactory results in online experiments. When the values are skewed towards zero, there is little promotion offered to the IMs. When a sizeable fraction of the dual variables has large values, we promote the IMs to the extent that disrupts the viewer experience. Therefore, the distribution of the dual values and their relationship to constraint violations provide essential insights to help us evaluate the re-ranking solution before we deploy it into the production system. We present a few representative plots from one of the runs in Fig. 3, 4, and 5, where the duals and the violations behave as expected.

Dual distribution: Generally, a high value of the dual implies that the corresponding constraint is difficult to satisfy. We prefer a distribution of duals that is not skewed towards zero or have inordinately high values. For reasons detailed in Section 7.2, we also desire a higher fraction of positive-valued duals. In Fig. 3, we refer to one such distribution (the dual values are clipped at the 95^{th} percentile for ease of illustration).

Dual distribution grouped by recommendation count: We also evaluate the distribution of the dual variables grouped by different segments of IMs based on the number of unique source members whose recommendation list the IMs appear in (see Fig. 4). These plots give us a good sense of which group of IMs benefits most from the LP formulation or whether the constraints need to be made tighter for IMs with higher recommendation counts. A higher recommendation count usually implies that the invitation constraint is easy to satisfy; hence the corresponding dual value must be close to zero. If the business requires the promotion of IMs with a higher recommendation count, the corresponding lower bound (ζ in formulation 2) must be increased.

Constraint violation: The constraint violation data (shown in Fig. 5) provides critical insights about the convergence and quality of the LP solution. The negative values in violation appear when the constraints are challenging to satisfy. The corresponding duals also assume positive values when the violations are negative. Ideally, the KKT condition implies a reversed L-shaped constraint violation plot, where the product of the dual and violation must be zero for all the constraints. However, in practice, achieving such a shape is often difficult. The convergence is better when the empirical plot looks similar to the reversed L shape.

7.1.2 Evaluating hits@top−k ratio. As mentioned in Section 4, the upper bound for the total measure of impressions made by the i^{th} source member δ_i penalizes sending out too many invitations. To understand the relationship between the hyper-parameter δ_i and the promotion of IMs in the recommender system, we simulate the hits@top−k ratio on the recommendation list with and without the boosting. This metric calculates the fraction of the IMs that appear in the top−k positions of the ranking results and provides a more intuitive estimation of the promotion offered. In section 7.2.5, we explain how we use this metric to tune δ_i to strike a balance between IM exposure and recommendation quality in production.

7.2 Online Experiment

7.2.1 Experiment Setup. As mentioned in Section 4, we add the duals (λ_j's) to the re-ranking formulation and modify the final relevance scores to promote certain IMs with personalized strength according to Eq. 5. This ranking solution is compared against the current scoring function of PYMK given in Eq. 1, which is the *baseline* we consider for all our online experiments. The practical business constraints elaborated in Section 5 force us to experiment with one treatment and control model. In future, we have plans to experiment with other non-LP-based solutions that might have more memory and compute requirements during the inference.

Metrics	*Group* 1	*Group* 2	*all IMs*
Invitations accepted	**+2.3%**	**+1.8%**	+0.1%
Sessions	+0.06%	+0.08%	+0.04%
Active Users	+0.06%	+0.03%	+0.01%

Table 1: Approach 1 Destination-side Results

Metrics	*Group* 1	*Group* 2
Invitations accepted	**+53.15%**	**+33.95%**
Sessions	**+5.88%**	**+3.88%**
Active Users	**+6.41%**	**+4.93%**

Table 2: Approach 1 Destination-side Results with Select IMs

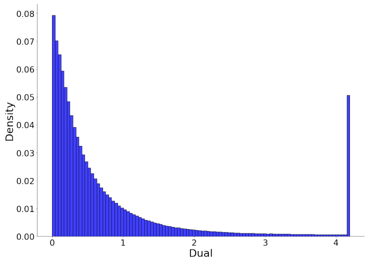

Figure 3: Distribution of Duals

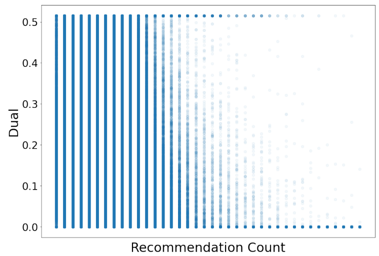

Figure 4: Duals vs Recommendation Count

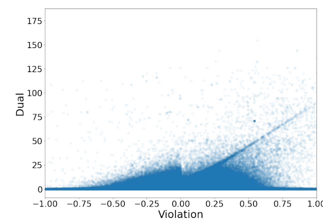

Figure 5: Duals vs Violations

Investing in such infrastructural changes is a multi-year effort. The proposed solution provides a practical trade-off to that end.

During the online A/B tests, we perform experiments on both sides, source and destination, to account for the violations of SUTVA in a two-sided marketplace [35]. To better understand the impact of our formulation, we further divide IMs into two groups based on their visiting frequency – *Group* 1 and *Group* 2. IMs from the first group visit the platform at least once in six months, while those from the second group visit once in three months.

When the candidates are ranked according to Eq. 1, IMs consistently receive lower final scores compared to active members and are ranked lower on average. In fact, the average SPR score of an IM is 2% lower than that of an active member (Group 1: 1.206, Group 2: 1.226, active members: 1.244), which results in only 8.8% of IMs in the top 10 recommendations. In the logged data collected according to the re-ranking formulation Eq. 1, around 11% of destination members (1.6% from Group 1, 8.5% from Group 2) are IMs.

7.2.2 Approach 1 *Results: Source-side Ramp with Aggressive Dual Accumulation.* Since in Approach 1, we cannot process many IMs in one instantiation of the LP solver; we adopt the dual accumulation strategy mentioned in Section 6.3, where we retain a dual value if an IM is assigned a value higher than predicted values from previous days. Unfortunately, such a strategy fails in the first source-side ramp, where we see a significant drop ($p < 0.001$) in invitation acceptance across all member segments.

Upon further investigation, we identify that such aggressive accumulation results in many of the IMs receiving exorbitantly high values of the duals, thereby overwhelming the recommendation list for all the source members. To explain the predicament further, after approximately 10 days of dual generation and accumulation, across all the source members, more than half of the IMs receive a relevance score higher than the maximum relevance score for the active members. This negative result conveys that overwhelming the recommendation list with IMs creates a negative experience among the viewers. Hence, they refuse to send invitations or visit the recommender system.

7.2.3 Approach 1 *Results: Destination-side Ramp.* To mitigate the overly aggressive aggregation, we opt for a milder accumulation scheme where only the first positive dual assigned to an IM is retained. Note that some form of dual accumulation is necessary to maintain the desired coverage during online tests.

Expectedly, the milder dual accumulation strategy results in statistically significant movement in invite-sent metrics for the IMs, which encourage us to launch the destination-side experiment. This experiment is kept alive for two weeks, upon the completion of which we observe the metrics reported in Table 1. Unless otherwise noted, the bold metrics have a p-value of less than 0.001, and non-bold metrics have a p-value of less than 0.05.

We anticipate that such small movement in the core metrics is due to the lower volume of the IMs that we can generate significant non-zero values of the duals for. To verify this conjecture, we recalculate the above statistics with only the IMs in the treatment group whose dual values are greater than the ones at the 99$^{\text{th}}$ percentile. We observe significant movement in these metrics as reported in Table 2. These results convey that the re-ranking strategy works only when a substantial fraction of the IMs receives enough promotion through the LP solution. Hence, scaling the LP formulation to more IMs (i.e. Approach 2) seems to be the only plausible recourse to a successful read for the destination-side experiment.

7.2.4 Approach 2 *Results: Source-side Ramp.* From our offline analysis, we understand the need to simplify the parametric form of the thresholds for the box-cut constraints (δ_i's). To that end, we choose $\delta_i = 5$ when i stands for an IM and $\delta_i = 10$ for an active member. This specific choice of the values of δ_i is guided by some statistics about the average number of destination members a source member gets to see in each session.

To our surprise, the first source-side ramp with this revised set-up gives an invitation-sent lift of as much as 900% for the IMs compared to the existing baseline. On the other hand, the invitation-accept rate for active members drops by as much as 30%. This magnitude of change is drastic and unacceptable for the overall health of the PYMK ecosystem. Unlike the complete antipathy of the source members to send out invitations that we observe in Section 7.2.2, we retain the interests and curiosity of the source members in this experiment. However, the balance is still overly shifted towards the IMs, so a better tuning of the δ_i's is necessary.

Groups	$\delta_i = 15$	$\delta_i = 18$	$\delta_i = 20$	$\delta_i = 25$
RecentDormants	**+53.87%**	**+28.72%**	**+38.10%**	**+22.54%**
One-by-Ones	**+39.10%**	**+19.51%**	**+26.19%**	**+15.03%**

Table 3: Approach 2 Source-side Invitation Sent Results

7.2.5 Approach 2 Results: Source-side Parallel Ramps. Our offline analyses with the hits@top−10 metrics provide some cues regarding the operating region of the δ_i's. Hits@top−10 hovers around 8.8% for the baseline re-ranking formulation. With the boost applied according to the λ_j's derived for different offline LP solution corresponding to $\delta_i \in \{15, 18, 20, 25\}$, hits@top−10 goes up by $\{17.3\%, 14.4\%, 10.6\%, 9.9\%\}$ respectively. One can see that a lower value of δ_i, on average, promotes a higher fraction of IMs in the top−10 positions. This offline metric provides a rather crude characterization of the online metrics. So to understand the implications of the variations in δ_i's, we ramp four parallel experiments with recommendations served according to the corresponding duals.

In the online experiments, the invitation-sent metrics do not change as monotonically as we observe in the offline hits@top−10 metrics (Table 3). However, the general trend still holds (i.e. the invitation-sent rate is the highest for the lowest value of δ_i and vice-versa). Since the aggregated invite-accepts become slightly negative for $\delta_i = 15$ and $\delta_i = 18$, we choose $\delta_i = 20$ as our operating point to maintain a good balance between IM exposure and acceptance of the invitations. This setting provides a healthy hike in invitations sent to the IMs without significantly altering the invitation-sent and accept rates for the active members.

7.2.6 Approach 2 Results: Destination-side Ramp. We ramp the LP formulation with $\delta_i = 20$ in the destination-side attention balancing experiment [35]. This framework provides a high-quality measurement of the impact of the treatment on the receivers of PYMK invitations. With frequent and relevant invitations sent out, we expect the IMs to engage more.

More specifically, the destination-side experiment is conceptualized by ramping the treatment to 25% of the source-side traffic and splitting the destination members into the control and treatment buckets. As Table 4 shows, IMs promoted through the treatment variant receive and accept more invitations. Quite intriguingly, the IMs who accept more invitations also start engaging with the platform, influencing the active user metrics in more than one way. For example, we resurrect[2] more IMs through the treatment, which suggests that our solution can target IMs that have the potential to be retained. More importantly, the IMs that we resurrect are also retained. After receiving the invitations, these IMs keep visiting the platform as Active Users in the subsequent days, which gets reflected in the significant increase in Retained IM Active Users metric. We also observe global movement in the unique member forming edges because of the promotion – +0.04% overall and +0.24% and +0.40% for Group 1 and Group 2, respectively.

Remarkably, the metrics improvement in the IM segment is complemented by a neutral trend in the aggregated metrics, such as the total invitations accepted, active users and sessions. Such results are pretty encouraging for the PYMK ecosystem. They demonstrate that careful reapportioning of the invitations in favour of long-term network health is possible without significantly affecting the experience of the active members who drive most of the above metrics. This propitious observation verifies our conjecture that IMs can be captivated by our platform with appropriate and relevant invitations. In future, we may categorize the IMs into different groups based on their likelihood of conversion and more methodically leverage their attention and interests. Currently, this system is ramped as a majority member experience whereby almost all of the 850 million members experience the re-ranking achieved by the constrained optimization formulation.

[2]If a member, who did not visit the platform for the past three months, suddenly visits the platform and subsequently becomes an active member within the next six months, we call that an event of "resurrection". Resurrected members emerge from dormancy and behave like active users within six months of receiving invitations.

Metrics	*Group* 1	*Group* 2	*All Members*
Invitations accepted	**+4.64%**	**+1.95%**	NS
IM Resurrected	+0.08%	+0.13%	+0.09%
Retained IM Active Users	+0.27%	+0.17%	+0.12%
Active Users	+0.14%	+0.06%	NS

Table 4: Approach 2 Destination-side Results

8 CONCLUSION

This paper prescribes a generic strategy to solve constrained multi-objective optimization problems using large-scale LP formulation. The paradigm we present is about matching entities of two different types at an enormous scale, which the re-rankers in modern-day recommender systems can be cast as. We present a complete recipe of how to design such re-rankers, with precise accommodation for practical business constraints and challenges.

In particular, we present an application where the IMs are systematically promoted in the recommendation list with the addition of a member-level constraint that encourages sending out more invitations to the IMs. The online experiments manifest a few characteristics that have been hitherto unknown:

- Promotion of IM conforms to certain balancing acts. If they are promoted heavily, they overwhelm the source members, and both sides' experience gets compromised. As a result, the source members develop extreme aversion towards sending out any invitation. A balanced exposure of the IMs intrigues the source members. Still, that inducement must be carefully counterpoised so that the active members, the primary drivers of the core business metrics, are not ignored. Finding the right operating region requires extensive offline evaluation and a few iterations of online experiments.
- The fraction of IMs who receive positive-valued duals significantly affects the magnitude of movement in the business metrics. Further investigations indicate that IMs who receive higher but controlled promotion engage more with the platform and generate more sessions. A better strategy to further categorize IMs based on their propensity to accept invitations and elevated engagement might benefit network health in the long run.

ACKNOWLEDGEMENT

Rahul Mazumder contributed to the work while he was a consultant for LinkedIn (in compliance with MIT's outside professional activities policies)

REFERENCES

[1] L. A. Adamic and B. A. Huberman. 2000. Power-Law Distribution of the World Wide Web. *Science* 287, 5461 (2000), 2115–2115.

[2] D. Agarwal, B. Chen, P. Elango, and X. Wang. 2012. Personalized Click Shaping through Lagrangian Duality for Online Recommendation. In *Proc. of SIGIR*. 485–494.

[3] D. Agarwal, B. Chen, R. Gupta, J. Hartman, Q. He, A. Iyer, S. Kolar, Y. Ma, P. Shivaswamy, A. Singh, and L. Zhang. 2014. Activity Ranking in LinkedIn Feed. In *Proc. of KDD*. 1603–1612.

[4] D. Agarwal, B. Chen, Q. He, Z. Hua, G. Lebanon, Y. Ma, P. Shivaswamy, H. Tseng, J. Yang, and L. Zhang. 2015. Personalizing LinkedIn Feed. In *Proc. of KDD*. 1651–1660.

[5] S. Agrawal and N. Goyal. 2013. Thompson Sampling for Contextual Bandits with Linear Payoffs. In *Proc. of ICML*. III–1220–III–1228.

[6] A. Anderson, D. P. Huttenlocher, J. M. Kleinberg, J. Leskovec, and M. Tiwari. 2015. Global Diffusion via Cascading Invitations: Structure, Growth, and Homophily. In *Proc. of WWW*. 66–76.

[7] E. M. Azevedo and E. G. Weyl. 2016. Matching markets in the digital age. *Science* 352, 6289 (2016), 1056–1057.

[8] K. Basu, A. Ghoting, R. Mazumder, and Y. Pan. 2020. ECLIPSE: An Extreme-Scale Linear Program Solver for Web-Applications. In *Proc. of ICML*. 704–714.

[9] D. Bertsekas. 1997. Nonlinear programming. *Journal of the Operational Research Society* 48, 3 (1997), 334–334.

[10] D. Bertsimas and J. Tsitsiklis. 1997. *Introduction to Linear Optimization* (1st ed.). Athena Scientific.

[11] R. E. Bixby. 2002. Solving Real-World Linear Programs: A Decade and More of Progress. *Oper. Res.* 50, 1 (feb 2002), 3–15.

[12] R. H. Byrd, P. Lu, J. Nocedal, and C. Zhu. 1995. A Limited Memory Algorithm for Bound Constrained Optimization. *SIAM Journal on Scientific Computing* 16, 5 (1995), 1190–1208.

[13] D. Chakrabarty, P. Jain, and P. Kothari. 2014. Provable Submodular Minimization using Wolfe's Algorithm. In *Proc. of Neurips*, Vol. 27.

[14] J. Denrell and G. Le Mens. 2017. Information Sampling, Belief Synchronization, and Collective Illusions. 63, 2 (feb 2017), 528–547.

[15] Q. Ding, Y. Liu, C. Miao, F. Cheng, and H. Tang. 2021. A Hybrid Bandit Framework for Diversified Recommendation. In *Proc. of AAAI*. 4036–4044.

[16] J. Duchi, S. Shalev-Shwartz, Y. Singer, and T. Chandra. 2008. Efficient projections onto the l1-ball for learning in high dimensions. In *Proc. of ICML*. 272–279.

[17] S. Falkner, A. Klein, and F. Hutter. 2018. BOHB: Robust and Efficient Hyperparameter Optimization at Scale. In *Proc. of ICML*. 1436–1445.

[18] J. L. Gearhart, K. L. Adair, J. D. Durfee, K. A. Jones, N. Martin, and R. J. Detry. 2013. *Comparison of open-source linear programming solvers.* Technical Report. Sandia National Lab.(SNL-NM), Albuquerque, NM (United States).

[19] F. Germano, V. Gómez, and G. Le Mens. 2019. The Few-Get-Richer: A Surprising Consequence of Popularity-Based Rankings?. In *Proc. of WWW*. 2764–2770.

[20] J. Gillenwater, A. Kulesza, and B. Taskar. 2012. Near-Optimal MAP Inference for Determinantal Point Processes. In *Proc. of NIPS*, Vol. 25. Curran Associates, Inc.

[21] J. Gondzio and R. Sarkissian. 2003. Parallel interior-point solver for structured linear programs. *Mathematical Programming* 96, 3 (2003), 561–584.

[22] R. Gupta, G. Liang, and R. Rosales. 2017. Optimizing Email Volume For Sitewide Engagement. In *Proc. of CIKM*. 1947–1955.

[23] R. Gupta, G. Liang, H. Tseng, R.K.H. Vijay, X. Chen, and R. Rosales. 2016. Email Volume Optimization at LinkedIn. In *Proc. of KDD*. 97–106.

[24] Gurobi Optimization, LLC. 2023. Gurobi Optimizer Reference Manual. https://www.gurobi.com

[25] W. Hamilton, Z. Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. In *Proc. of NIPS*. 1024–1034.

[26] G. W. Imbens and D. B. Rubin. 2015. *Causal Inference for Statistics, Social, and Biomedical Sciences: An Introduction.* Cambridge University Press.

[27] T. Joachims, L. Granka, B. Pan, H. Hembrooke, F. Radlinski, and G. Gay. 2007. Evaluating the Accuracy of Implicit Feedback from Clicks and Query Reformulations in Web Search. *ACM Trans. Inf. Syst.* 25, 2 (apr 2007), 7–es.

[28] T. Joachims, A. Swaminathan, and T. Schnabel. 2018. Unbiased Learning-to-Rank with Biased Feedback. In *Proc. of IJCAI*. 5284–5288.

[29] N. Kallus, X. Mao, K. Wang, and Z. Zhou. 2022. Doubly Robust Distributionally Robust Off-Policy Evaluation and Learning. In *Proc. of ICML (Proceedings of Machine Learning Research, Vol. 162)*. PMLR, 10598–10632.

[30] H. Kiyohara, Y. Saito, T. Matsuhiro, Y. Narita, N. Shimizu, and Y. Yamamoto. 2022. Doubly Robust Off-Policy Evaluation for Ranking Policies under the Cascade Behavior Model. In *Proc. of WSDM*. ACM.

[31] G. Le Mens, J. Denrell, B. Kovács, and H. Karaman. 2019. Information sampling, judgment, and the environment: Application to the effect of popularity on evaluations. *Topics in cognitive science* 11, 2 (2019), 358–373.

[32] P. Lee, L. V.S. Lakshmanan, M. Tiwari, and S. Shah. 2014. Modeling Impression Discounting in Large-Scale Recommender Systems. In *Proc. of KDD*. 1837–1846.

[33] C. Lo, E. Longueau, A. Saha, and S. Chatterjee. 2020. Edge formation in Social Networks to Nurture Content Creators. In *Proc. of WWW*. 1999–2008.

[34] Robert K Merton. 1968. The Matthew Effect in Science: The reward and communication systems of science are considered. *Science* 159, 3810 (1968), 56–63.

[35] P. Nandy, K. Basu, S. Chatterjee, and Y. Tu. 2020. A/B Testing in Dense Large-Scale Networks: Design and Inference. In *Proc. of NeurIPS*, H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin (Eds.), Vol. 33. Curran Associates, Inc., 2870–2880.

[36] A. Nigam, P. Agrawal, and A. Jain. 2020. Recommending network connections by optimizing for two-sided implicit value of an edge. https://patents.justia.com/patent/11526786

[37] R. F. Prudencio, M. R. O. A. Maximo, and E. L. Colombini. 2022. A Survey on Offline Reinforcement Learning: Taxonomy, Review, and Open Problems. https://doi.org/10.48550/ARXIV.2203.01387

[38] R. Ramanath, S. S. Keerthi, K. Basu, K. Salomatin, and Y. Pan. 2021. Efficient Algorithms for Global Inference in Internet Marketplaces. *arXiv preprint arXiv:2103.05277*. https://arxiv.org/abs/2103.05277

[39] R. Ramanath, S. S. Keerthi, K. Basu, K. Salomatin, P. Yao, and A. Ghoting. 2021. DuaLip: Dual Decomposition based Linear Program Solver, version 1.0.0. https://github.com/linkedin/dualip

[40] D. B. Rubin. 2005. Causal Inference Using Potential Outcomes. *J. Amer. Statist. Assoc.* 100, 469 (2005), 322–331.

[41] M. J. Salganik, P. S. Dodds, and D. J. Watts. 2006. Experimental study of inequality and unpredictability in an artificial cultural market. *science* 311, 5762 (2006), 854–856.

[42] C. Sha, X. Wu, and J. Niu. 2016. A Framework for Recommending Relevant and Diverse Items. In *Proc. of IJCAI*. 3868–3874.

[43] Y. Shi, M. Kim, S. Chatterjee, M. Tiwari, S. Ghosh, and R. Rosales. 2016. Dynamics of Large Multi-View Social Networks: Synergy, Cannibalization and Cross-View Interplay. In *Proc. of KDD*. 1855–1864.

[44] A. Slivkins et al. 2019. Introduction to multi-armed bandits. *Foundations and Trends® in Machine Learning* 12, 1-2 (2019), 1–286.

[45] W. Su, S. Boyd, and E. Candes. 2014. A Differential Equation for Modeling Nesterov's Accelerated Gradient Method: Theory and Insights. In *Proc. of NIPS*, Vol. 27.

[46] Y. Tu, C. Lo, Y. Yuan, and S. Chatterjee. 2019. Feedback Shaping: A Modeling Approach to Nurture Content Creation. In *Proc. of KDD*. ACM, 2241–2250.

[47] M. J. P. Walker, B. Yan, Y. Xiao, Y. Wang, and A. Acharya. 2020. Isometric Graph Neural Networks. *CoRR* abs/2006.09554 (2020). https://arxiv.org/abs/2006.09554

[48] Q. Wu, Y. Liu, C. Miao, Y. Zhao, L. Guan, and H. Tang. 2019. Recent Advances in Diversified Recommendation. *CoRR* abs/1905.06589 (2019). arXiv:1905.06589 http://arxiv.org/abs/1905.06589

[49] Y. Yuan, J. Zhang, S. Chatterjee, S. Yu, and R. Rosales. 2019. A State Transition Model for Mobile Notifications via Survival Analysis. In *Proc. of WSDM*. 123–131.

[50] H. Zheng and J. Wu. 2017. Online to offline business: urban taxi dispatching with passenger-driver matching stability. In *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 816–825.