Nested Logit Choice Model of the Inequity Aversion Pricing Problem on Social Networks

Sunil Chopra
Kellogg School of Management, Northwestern University, Evanston, IL 60208, USA, s-chopra@kellogg.northwestern.edu

Hyunwoo Park
Graduate School of Data Science, Seoul National University, Seoul 08826, Republic of Korea, hyunwoopark@snu.ac.kr

Sangho Shim
School of Engineering, Mathematics and Science, Robert Morris University, Moon Twp, PA 15108, USA, shim@rmu.edu

The inequity aversion pricing problem, which maximizes total revenue, involves offering not-too-different discount prices to friends in a social network. In this paper, we introduce a choice model of the inequity aversion pricing problem using a more realistic customer choice rule called nested logit and two solution methods: exact and approximate solution methods. For an exact solution, we develop a linear formulation of the nested logit choice model of the inequity aversion pricing problem and enhance the exact solution method by bounding the variables of choice probabilities and adding only important cutting planes. We then develop a heuristic algorithm to solve this problem in large social networks.

Key words: inequity aversion pricing, economic choice model, social networks, independent set, conflict graph, revenue management

OR/MS subject classification: Network/graphs, Promotion

---

1. Introduction

The inequity aversion pricing problem that maximizes total revenue is the problem of offering discount prices that are not too different for friends on a social network in which the nodes and the edges represent the potential customers and the friendship relations. On the social network, a conflict graph can be constructed to represent the pairs of too different options by edges, and an independent set of the conflict graph represents a feasible offer of discount price options to the customers. To solve this problem, a value function has been used to define the attractiveness of the discount price options to the potential customer in the social network (Fehr and Schmidt 1999, Bolton et al. 2000, Alon et al. 2013, Amanatidis et al. 2019, Chopra et al. 2022): If the discount price is less than the value of the product to the customer, the customer buys the product, and the discount price is the revenue generated by the offer. In practice, a value function is often not available. Still, we can use historical data and external approaches, such as customer surveys, to estimate the revenue we expect the offer to generate. Instead of the value function, a choice model can estimate the expected revenue by the choice probability of a customer accepting the offer and choosing to purchase the product, called redeeming the offered discount price option. In this paper, we introduce a choice model of the inequity aversion pricing problem.

The most commonly used rule for calculating choice probabilities is the multinomial logit (MNL) rule, which assumes that there is no correlation between the available alternatives (including the non-purchase alternative in case of the inequity aversion pricing problem). However, customers actually make decisions sequentially, such as whether to upgrade their smartphone and then which product to choose. Between similar products to purchase, there should be some positive correlations. The nested logit (NL) rule accounts for sequential decisions by grouping alternatives into subsets called nests. When offering discount price options, the company may consider two nests for the customer: a singleton nest of no purchase and a nest of redeeming one of the discount price options offered to the customer for similar products of the company.

We develop a mixed integer linear program (MILP) for the NL model of the inequality aversion pricing problem using the outer approximation cuts introduced in Ljubíc and Moreno (2018). Although an exponentially large number of outer approximation cuts define the entire MILP, a branch-and-cut procedure can efficiently solve it on medium-scale social networks. To tackle the large-scale inequality aversion pricing problem, we perform multiple iterations of a heuristic approach simultaneously, beginning with multiple initial solutions.

Section 2 introduces an NL model of the inequity aversion pricing problem. The model is formulated as a mixed integer nonlinear program followed by a linear reformulation (mixed integer linear program or MILP) which can be optimally solved by integer linear programming solvers such as GUROBI and CPLEX. To solve the MILP on medium-scale social networks quickly, Section 3.1 enhances the MILP by adding constraints that bound the decision variables of choice probabilities. Adding only important cutting planes also speeds up the MILP significantly. To tackle a large-scale inequity aversion pricing problem, Section 3.2 develops a heuristic approach, which is a local search method that starts from an initial feasible solution. A parallel machine can perform multiple iterations of the heuristic on multiple initial solutions, called single and double options. In Section 4, computational experiments verify that the proposed solution methods identify optimal or near-optimal solutions to the problem efficiently. Our heuristic algorithm tackles the NL choice model of the inequity aversion pricing problem on two large-scale social networks of 200,000 and 1,000,000 customers. Section 5 concludes the results of this paper and suggests future research.

2. Problem Formulations

We represent a social network as an undirected graph $G = (V, E)$, with $|V| = n$. The nodes correspond to potential customers and the edges correspond to the pairs of customers who are socially connected. Providers (retailers) of some goods or services have a limited number of discount price options that they can offer to their customers. Each customer is assumed to redeem one of the options offered or not purchase the goods or service. Let $\mathcal{J}$ denote the set of all possible *discount price options* (or simply *options*) from which the provider selects options to be offered to each customer. The provider may offer the customer multiple options across multiple products.

A *revenue function* $r : \mathcal{J} \to \mathbb{R}$ maps an option $j \in \mathcal{J}$ to its price $r(j)$: The revenue from offering a discount price option is the price offered, and multiplying this revenue by the customer's probability of purchase is the expected revenue that the provider can earn by offering the discount price option. A choice model for the inequity averse pricing problem finds the discount price offer that maximizes total expected revenue for all customers. The options available to each customer and friend pair must not be too much different. In particular, at most one discount price option can be offered to each customer and friend pair for each product. In this work, each customer is assumed to redeem at most one option from among the available ones offered by the retailer.

We employ a binary vector $Y = (Y(i,j) \in \{0,1\} : i \in V, j \in \mathcal{J})$ that indicates discount option j of price $r(j)$ offered to customer $i \in V$. Too different prices are not offered to a customer or to friends for each product. Those pairs of offers $(i,j),(h,k) \in V \times \mathcal{J}$ that are not tolerable are captured by the edges $(i,j) \sim (h,k) \in \hat{E}$ (or denoted by $\{(i,j),(h,k)\} \in \hat{E}$ interchangeably) of the conflict graph $\hat{G} = \left(\hat{V} = V \times \mathcal{J}, \hat{E}\right)$. For the conflict graph edge $(i,j) \sim (h,k)$, at most one of the two options can be offered, which is ensured by

$$Y(i,j) + Y(h,k) \le 1 \text{ for } (i,j) \sim (h,k) \in \hat{E}.$$

That is, a feasible offer $Y \in \mathcal{Y}^{\mathrm{STB}}$ is an independent set (also called stable set) of the conflict graph, where $\mathcal{Y}^{\mathrm{STB}}$ denotes the set of independent sets (or stable sets) in the conflict graph.

Figure 1 is a conflict graph $\hat{G}$ (in black) defined on a social network $G = (V, E)$ (in gray at the bottom) with three customers $V = \{u, v, w\}$ and two friendship relations $E = \{u \sim v, v \sim w\}$. All possible options are $\mathcal{J} = \{1,2,3,4,5,6\}$ for 3 versions (products) of smart phones (Smart Phone 1, Smart Phone 2, Smart Phone 3). The options are associated with revenue function $r(1) = \$100, ..., r(6) = \$600$. The oldest version (Smartphone 1) might be heavily discounted to \$100, while the newest version (Smartphone 3) might have little to no discount. The graph (in gray) on the left ensures that two different options are not offered simultaneously for the same version. Multiple options may be offered to a customer, but she purchases at most one smart phone. In fact, the conflict graph may have a customer-specific structure within and between customers, and this paper introduces solution methods for general conflict graphs.

Given a feasible offer $Y \in \mathcal{Y}^{\mathrm{STB}}$, $P_Y(i,j)$ denotes the choice probability of i redeeming j. A choice model of this multi-product inequity aversion pricing problem is

$$\max \left\{ \sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{J}_i^Y} r(j) P_Y(i,j) : Y \text{ is a feasible offer} \right\},$$

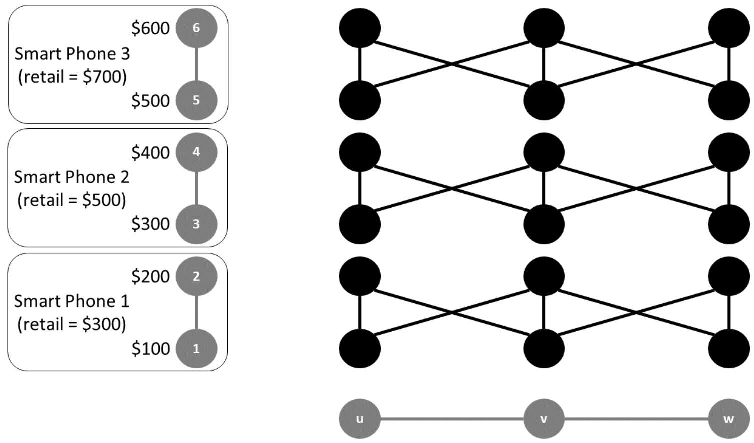

Figure 1 **A conflict graph on a social network of three customers $V = \{u, v, w\}$ with 6 options of 3 products**

where $\mathcal{J}_i^Y = \{j \in \mathcal{J} : Y(i,j) = 1\}$ is the set of the options offered to i. The probabilities $P_Y(i,j)$ depend on how we define the customer choice rule on the feasible offer Y. For a better understanding, before we describe NL rules in Section 2.2, we first discuss the basic concepts of MNL rules in the context of offering the optimal discount prices in Section 2.1. In our choice model, we assume that every customer $i \in \mathcal{I}$ always has a choice of no-purchase with non-zero probability. The additional option of no-purchase is referred to as the base option, denoted by $j = 0$. Since the base case is always in the choice set with non-zero probability, both $\mathcal{J}_i^Y \subset \mathcal{J}$ and $\mathcal{J}_i^Y \cup \{j = 0\}$ are referred to as the choice set for customer i without confusion. Note that $\sum_{j \in \mathcal{J}_i^Y} P_Y(i,j) = 1 - P_Y(i,0) < 1$.

2.1. Multinomial Logit Model

Logit models allow us to perform regression analysis between the choice probability and certain features such as the discount rate and the newness of the product. For example, consider a regression analysis between the discount price $r(j)$. Because a probability is limited between 0 and 1, we do not perform linear regression directly between $r(j)$ and $P(i,j)$. Instead, we consider the ratio of the choice probability of the option to the probability of the base option, called odds:

$$\mathrm{odds}(i,j) = \frac{P(i,j)}{P(i,0)} \text{ for } j \in \mathcal{J}.$$

We may perform a linear regression between the choice probabilities and the log-odds:

$$\log \mathrm{odds}(i,j) = \log \frac{P(i,j)}{P(i,0)} = \beta_0(i) + \beta_1(i) r(j)$$

where $\beta_0(i)$ and $\beta_1(i)$ denote regression coefficients. The value of j to i is measured by utility $V(i,j)$ defined by

$$\frac{P(i,j)}{P(i,0)} = \frac{e^{V(i,j)}}{e^{V(i,0)}} = e^{\beta_0(i) + \beta_1(i) r(j)}.$$

The utility $V(i,j) = \beta_0(i) + \beta_1(i)r(j)$ measures the value of each option j to the customer i. By subtracting the base utility $V(i,0)$ from the utilities $V(i,j)$ for all $j \in \{0\} \cup \mathcal{J}$, we set the utilities to $V(i,0) = 0$ and $V(i,j) = \beta_0(i) + \beta_1(i)r(j)$. For this feature of discount price $r(j)$, the choice probability $P(i,j)$ should be negatively correlated (*i.e.*, $\beta_1(i) < 0$).

A multinomial logit (MNL) choice model assumes the Independence of Irrelevant Alternatives (IIA) condition. It assumes that the odds of an alternative does not change by the other alternatives, and allows us to compute the odds under a simple assumption. Let $q(i,j)$ denote the choice probability assuming that $j \in \mathcal{J}$ is the only option that is offered to i; *i.e.*, $\mathcal{J}_i = \{j\}$. For any choice set $\mathcal{J}_i \subseteq \mathcal{J}$, the odds of $j \in \mathcal{J}_i$ is

$$\text{odds}(i,j) := \frac{P(i,j)}{P(i,0)} = \frac{q(i,j)}{q(i,0)} = \frac{q(i,j)}{1-q(i,j)} = e^{V(i,j)}. \tag{1}$$

The choice probabilities $P(i,j) = P_Y(i,j)$ may be different across the choice sets indicated by Y; *i.e.*, $Y \neq Y'$ may imply $P_Y(i,j) \neq P_{Y'}(i,j)$ where $Y \neq Y'$ indicate the different choice sets $\mathcal{J}_i \neq \mathcal{J}_i'$ for individual i. Note that the odds are same by the IIA property if the option belongs to both $\mathcal{J}_i$ and $\mathcal{J}_i'$: *i.e.*, if $j \in \mathcal{J}_i \cap \mathcal{J}_i'$ (or $Y(i,j) = Y'(i,j) = 1$), then

$$\text{odds}(i,j) = \frac{q(i,j)}{1-q(i,j)} = \frac{P_Y(i,j)}{P_Y(i,0)} = \frac{P_{Y'}(i,j)}{P_{Y'}(i,0)}.$$

A mixed-integer non-linear program of the multinomial logit model of the inequity aversion pricing problem is

$$\text{(MINLP-MNL)} \quad \max \quad \sum_{i\in\mathcal{I}} \sum_{j\in\mathcal{J}} r(j)P(i,j) \tag{2}$$

$$\text{subject to} \quad Y(i,j) + Y(h,k) \le 1 \text{ for } (i,j) \sim (h,k) \in \hat{E} \tag{3}$$

$$P(i,j) = \frac{e^{V(i,j)}Y(i,j)}{1 + \sum_{k\in\mathcal{J}} e^{V(i,k)}Y(i,k)} = \frac{\text{odds}(i,j)Y(i,j)}{1 + \sum_{k\in\mathcal{J}} \text{odds}(i,k)Y(i,k)} \text{ for } i \in \mathcal{I}, j \in \mathcal{J}. \tag{4}$$

Given a chocie set $\mathcal{J}_i = \{j \in \mathcal{J} : Y(i,j) = 1\}$, McFadden (1974) first showed the representation (4) of the choice probabilities. The choice probabilities $P(i,j)$ are determined by Y and we denote the objective function in (2) by

$$\text{REVENUE}(Y) = \sum_{i\in\mathcal{I}} \sum_{j\in\mathcal{J}} r(j)P(i,j) \tag{5}$$

which is the expected total revenue from the discount price offer indicated by Y. Note that the optimal solution Y^* does not have to indicate a maximal independent set. For example, assume that $\mathcal{J}_i = \{1,6\}$, $\mathcal{J}_i' = \{6\}$ and $\text{odds}(i,1) = \text{odds}(i,6) = 1$. Then, $P(i,0) = P(i,1) = P(i,6) = 1/3$ and $P'(i,0) = P'(i,6) = 1/2$ for $\mathcal{J}_i$ and $\mathcal{J}_i'$, and the revenue from $\mathcal{J}_i = \{1,6\}$ is less than that from $\mathcal{J}_i' = \{6\}$:

$$r(1)P(i,1) + r(6)P(i,6) = \frac{100}{3} + \frac{600}{3} < \frac{600}{2} = r(6)P'(i,6).$$

The IIA property linearizes the (MINLP-MNL) to a MILP:

$$\text{(MILP-MNL)} \quad \max \sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{J}} r(j)P(i,j) \tag{6}$$

$$\text{subject to} \quad Y(i,j)+Y(i',j')\le 1 \text{ for } \{(i,j),(i',j')\}\in\hat{E} \tag{7}$$

$$P(i,j)\le Y(i,j) \text{ for } i\in\mathcal{I} \text{ and } j\in\mathcal{J} \tag{8}$$

$$P(i,0)+\sum_{j\in\mathcal{J}} P(i,j)=1 \text{ for } i\in\mathcal{I} \tag{9}$$

$$P(i,j)\le \text{odds}(i,j)P(i,0) \text{ for } i\in\mathcal{I} \text{ and } j\in\mathcal{J} \tag{10}$$

$$P(i,0)\le \frac{1}{\text{odds}(i,j)}P(i,j)+(1-Y(i,j)) \tag{11}$$

Note that (8), (10) and (11) ensure the IIA property; *i.e.*,

$$\frac{P(i,j)}{P(i,0)}=\text{odds}(i,j)Y(i,j) \text{ for } i\in\mathcal{I}, j\in\mathcal{J}.$$

The MNL model is a random utility maximization (RUM) model, in which the utilities $V(i,j)$ are systematic components of random utilities

$$U(i,j)=V(i,j)+\varepsilon(i,j) \tag{12}$$

where $\varepsilon(i,j)$ are error components. The choice probability $P(i,j)$ that individual i chooses alterative j is equal to that of $U(i,j)$ being the largest of all $U(i,k)$ for $k\in\mathcal{J}_i$:

$$\begin{aligned} P(i,j) &= P\left[U(i,j)\ge U(i,k) \text{ for all } k\in\mathcal{J}_i\right] \\ &= P\left[\varepsilon(i,k)-\varepsilon(i,j)\le V(i,j)-V(i,k) \text{ for all } k\in\mathcal{J}_i\right] \end{aligned}$$

Across all $j\in\mathcal{J}_i$, the MNL model assumes that $\varepsilon(i,j)$ are i.i.d. as Extreme Value Type I. For more details of RUM models, the readers may refer to Heiss (2002).

2.2. Nested Logit Model

The nested logit (NL) model can be derived from a RUM model by generalizing the MNL model where the error terms $\varepsilon(i,j)$ are i.i.d. as Extreme Value Type I. Instead, the NL model assumes a generalized version of this distribution. This special form of the generalized extreme value (GEV) distribution extends the Extreme Value Type I distribution by allowing the alternatives within a nest to have mutually correlated error terms.

In this study, we consider two nests $\mathcal{N}$: a nest for buying no product (*i.e.*, $\mathcal{N}=\{j=0\}$), and the other nest for the offered discount price options (*i.e.*, $\mathcal{N}=\mathcal{J}^Y$ or $\mathcal{N}=\mathcal{J}$ with $P(i,j)=0$ for $j\in\mathcal{J}\setminus\mathcal{J}^Y$). The joint distribution of the error terms has an additional parameter $\lambda=\sqrt{1-\rho}$, called log-sum parameter or dissimilarity parameter interchangeably in this paper, that represents a measure of the mutual correlation of the error terms of all discount price options, where ρ represents the correlation coefficient. The log-sum parameter is an inverse measure of the correlation. The marginal distribution of each error term is again Extreme Value Type I.

In the NL model, the probability of selecting an alternative $j \in \{0\} \cup \mathcal{J}$ is expressed as the product of the probability of selecting the related nest $\mathcal{N}$, and the conditional probability of selecting the alternative $j \in \mathcal{N}$ in the nest:

$$P[i \text{ selects } j] = P[i \text{ selects } \mathcal{N}] \times P[i \text{ selects } j \in \mathcal{N} | i \text{ selects } \mathcal{N}].$$

They are defined by

$$P(i,0) = \frac{e^{V(i,0)}}{e^{V(i,0)} + e^{\lambda \Gamma_i}} = \frac{1}{1 + e^{\lambda \Gamma_i}} \quad \text{for } i \in \mathcal{I} \tag{13}$$

$$\begin{aligned} P(i,j) &= \frac{e^{\lambda \Gamma_i}}{e^{V(i,0)} + e^{\lambda \Gamma_i}} \frac{e^{V(i,j)/\lambda} Y(i,j)}{\sum_{k \in \mathcal{J}} e^{V_{ik}/\lambda} Y(i,k)} \\ &= \frac{e^{\lambda \Gamma_i}}{1 + e^{\lambda \Gamma_i}} \frac{e^{V(i,j)/\lambda} Y(i,j)}{\sum_{k \in \mathcal{J}} e^{V_{ik}/\lambda} Y(i,k)} \quad \text{for } i \in \mathcal{I} \text{ and } j \in \mathcal{J} \end{aligned} \tag{14}$$

where $0 < \lambda \leq 1$ and

$$\Gamma_i = \ln \sum_{j \in \mathcal{J}} e^{V(i,j)/\lambda} Y(i,j) \text{ for } i \in \mathcal{I}. \tag{15}$$

Throughout this paper, we assume a positive correlation across the discount price options (*i.e.*, $0 \leq \rho < 1$ or $0 < \lambda \leq 1$) as they apply for similar products.

A mixed-integer non-linear program of a nested logit model of the inequity aversion pricing problem is

$$\text{(MINLP-NL}(\lambda)) \quad \max \left\{ \sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{J}} r(j) P(i,j) : (3) \text{ and } (14) \right\}. \tag{16}$$

The NL choice probabilities (14) are determined by Y, and the expected total revenue is a function of Y, denoted by

$$\text{REVENUE}^{\lambda}(Y) = \sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{J}} r(j) P_Y(i,j), \tag{17}$$

where $P_Y(i,j)$ are defined by (14). The NL choice model of the inequity aversion pricing problem can be simply written as

$$\max \left\{ \text{REVENUE}^{\lambda}(Y) : Y \in \mathcal{Y}^{\text{STB}} \right\},$$

where $\mathcal{Y}^{\text{STB}}$ is the set of feasible offers Y indicating the independent sets (also called stable sets interchangeably). In particular, if $\lambda = 1$ (or $\rho = 0$), the NL model is the MNL model, and the expected total revenue is

$$\text{REVENUE}^{\lambda}(Y) = \text{REVENUE}(Y)$$

where $\text{REVENUE}(Y)$ denotes the total revenue (5) of the MNL model.

Let $0 < \lambda < 1$ and let $\hat{w} : \mathbb{R}_+ \to \mathbb{R}$ be a differentiable function defined by

$$\hat{w}(z) = \frac{z^{\lambda}}{1 + z^{\lambda}}. \tag{18}$$

Then, its derivative is

$$\frac{d}{dz}\hat{w}(z) = \frac{\lambda z^{\lambda-1}}{\left(1 + z^{\lambda}\right)^2}. \tag{19}$$

It is easy to see that $\hat{w}$ is a concave increasing function. Let $Y_i = (Y(i,j) \in [0,1] : j \in \mathcal{J})$. For $i \in \mathcal{I}$, we define

$$z(Y_i) = \sum_{j \in \mathcal{J}} e^{V(i,j)/\lambda} Y(i,j).$$

Let $\mathcal{Y}_i^{\text{STB}}$ be the set of indicating vectors (denoted by $Y_i^{\text{STB}} = (Y^{\text{STB}}(i,j) \in \{0,1\} : j \in \mathcal{J})$) of independent sets in the subgraph $\hat{G}\left[\hat{V}_i\right]$ induced by $\hat{V}_i = \{(i,j) : j \in \mathcal{J}\} \subset \hat{V}$. Since $\hat{w}$ is a concave function, the value of $\hat{w}(z(Y_i))$ is bounded from above by its first-order approximation on a non-empty independent set $Y_i^{\text{STB}} \neq \mathbf{0} \in \mathcal{Y}_i^{\text{STB}} \subset \{0,1\}^{\mathcal{J}}$; *i.e.*,

$$\begin{aligned}
&\sum_{j \in \mathcal{J}} P(i,j) = \hat{w}(z(Y_i)) \\
&\le \hat{w}\left(z\left(Y_i^{\text{STB}}\right)\right) + \sum_{j \in \mathcal{J}} \frac{\partial \hat{w}}{\partial Y(i,j)}\left(z\left(Y_i^{\text{STB}}\right)\right) \cdot \left(Y(i,j) - Y^{\text{STB}}(i,j)\right) \\
&= \hat{w}\left(z\left(Y_i^{\text{STB}}\right)\right) + \sum_{j \in \mathcal{J}} \hat{w}'\left(z\left(Y_i^{\text{STB}}\right)\right) \cdot e^{V(i,j)/\lambda} \cdot \left(Y(i,j) - Y^{\text{STB}}(i,j)\right).
\end{aligned} \tag{20}$$

If $Y_i^{\text{STB}} = \mathbf{0}$, Constraint (8) will imply $P(i,j) = Y(i,j) = 0$ for all $j \in \mathcal{J}$ (*i.e.*, no options are offered to the customer and he will not redeem any options).

Equivalently, the valid inequality (20) can be rewritten as

$$\begin{aligned}
&\sum_{j \in \mathcal{J}} P(i,j) - \sum_{j \in \mathcal{J}} \hat{w}'\left(z\left(Y_i^{\text{STB}}\right)\right) \cdot e^{V(i,j)/\lambda} \cdot Y(i,j) \\
&\le \hat{w}\left(z\left(Y_i^{\text{STB}}\right)\right) - \sum_{j \in \mathcal{J}} \hat{w}'\left(z\left(Y_i^{\text{STB}}\right)\right) \cdot e^{V(i,j)/\lambda} \cdot Y^{\text{STB}}(i,j) \text{ for } i \in \mathcal{I} \text{ and } Y_i^{\text{STB}} \in \mathcal{Y}_i^{\text{STB}},
\end{aligned} \tag{21}$$

which we referred to as *outer approximation cuts.*

For $i \in \mathcal{I}$ and $Y_i^{\text{STB}} \neq \mathbf{0} \in \mathcal{Y}_i^{\text{STB}}$, a lower-bound is given by

$$\sum_{j\in\mathcal{J}} P(i,j) \geq \hat{w}\left(z\left(Y_i^{\text{STB}}\right)\right) - \sum_{j:Y^{\text{STB}}(i,j)=1} (1-Y(i,j)) - \sum_{j:Y^{\text{STB}}(i,j)=0} Y(i,j). \tag{22}$$

Note that $\hat{w}(z) < 1$ and the right-hand-side of (22) is negative whenever $Y_i = (Y(i,j) : j \in \mathcal{J}) \neq Y_i^{\text{STB}} \in \mathcal{Y}_i^{\text{STB}}$. Along with the outer approximation cuts (21), the lower bounding constraints (22) ensure total probability of purchase $\sum_{j\in\mathcal{J}} P(i,j) = 1 - P(i,0)$ and the base case probability $P(i,0)$ in (13). Then, Equations (14) are ensured by

$$P(i,j) \leq \frac{e^{V(i,j)/\lambda}}{e^{V(i,k)/\lambda}} P(i,k) + (1 - Y(i,k)) \text{ for } i \in \mathcal{I} \text{ and } (j,k) \in \mathcal{J} \times \mathcal{J} \text{ with } j \neq k, \tag{23}$$

across the available options j indicated by $Y(i,j) = 1$.

In (6)-(11), a MILP of NL model replaces (10)-(11) by (21)-(23):

$$(\text{MILP-NL}(\lambda)) \quad \max\left\{\sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{J}} r(j)P(i,j) : (7)\text{-}(9) \text{ and } (21)\text{-}(23)\right\}. \tag{24}$$

The MILP of NL model has exponentially many constraints (21)-(22), but a branch-and-cut procedure may solve this MILP efficiently.

3. Exact and Approximate Solution Methods

In this section, we introduce an exact solution method to optimally solve medium-scale instances and a heuristic algorithm to tackle large-scale instances. Although MILP-NL(λ) is defined by exponentially many constraints (21)-(22), a branch-and-cut procedure efficiently adds only necessary constraints to define feasible solutions close to the optimal solution. In Section 3.1, we speed up the branch-and-cut (B&C) procedure significantly by adding strong constraints that bound the variables of choice probabilities. We further speed up the B&C procedure by adding only important cutting planes. In Section 3.2, we introduce a local search algorithm to tackle large-scale instances. Beginning with multiple initial solutions, a parallel process can speed up the heuristic algorithm.

3.1. Enhancing the Branch-and-Cut Procedure

In the default setting of a solver (*e.g.*, GUROBI), the branch-and-cut procedure does not solve MILP-NL(λ) in (24) very efficiently. The procedure can be accelerated by bounding the decision variables of choice probabilities (*i.e.*, $P(i,j)$ for $i \in V$ and $j \in \mathcal{J}$) and adding only important cutting planes. The MILP (24) enhanced by the bounding constraints and the important cutting planes will be denoted by Enhanced(λ). In the next section (Section 4), numerical experiments will verify that the enhanced MILP can be solved much faster than the default setting of the MILP (24).

To develop the bounds, we associate the conflict graph nodes $(i,j) \in \hat{V}$ with node weight $e^{V(i,j)/\lambda}$ and denote by $\hat{G}_i = \hat{G}\left[\hat{V}_i\right]$ the subgraphs induced by $\hat{V}_i = \{(i,j) \in \hat{V} : j \in \mathcal{J}\}$ for the potential customers $i \in V$. For $(i,j) \in \hat{V}$, let $\hat{V}_i^{\max(j)} \subset \hat{V}_i$ be the maximum weight independent set including (i,j) in $\hat{G}_i$ and denote the weight of $\hat{V}_i^{\max(j)}$ by

$$z_i^{\max(j)} = \sum_{(i,k) \in \hat{V}_i^{\max(j)}} e^{V(i,k)/\lambda} \tag{25}$$

It is the upper bound of $z(Y_i^{\mathrm{STB}}) = \sum_{k \in \mathcal{J}} e^{V(i,k)/\lambda} Y^{\mathrm{STB}}(i,k)$ for all independent sets $Y_i^{\mathrm{STB}} \in \mathcal{Y}_i^{\mathrm{STB}}$ of the subgraph $\hat{G}_i = \hat{G}\left[\hat{V}_i\right]$ with $Y^{\mathrm{STB}}(i,j) = 1$ (*i.e.*, including (i,j)). In general, the maximum weight independent set problem is NP-hard. However, the number $|\mathcal{J}|$ of options is not too large to run exact solution methods (*e.g.*, brute-force search) for the maximum weight independent set problem. In particular, if $\hat{G}_i^j = \hat{G}_i \setminus \{(i,j)\}$ is a bipartite graph (*e.g.*, $\hat{G}_i$ are bipartite graphs in Figure 1), the maximum weight independent set problem on $\hat{G}_i^j$ can be solved in a polynomial time. The minimum weight independent set including (i,j) is trivially the singleton independent set $\hat{V}_i^{\min(j)} = \{(i,j)\}$ and we denote the weight of $\hat{V}_i^{\min(j)}$ by

$$z_i^{\min(j)} = e^{V(i,j)/\lambda}, \tag{26}$$

which is the lower bound of $z(Y_i^{\mathrm{STB}}) = \sum_{k \in \mathcal{J}} e^{V(i,k)/\lambda} Y^{\mathrm{STB}}(i,k)$ with $Y^{\mathrm{STB}}(i,j) = 1$.

Since $\hat{w}(z)$ is an increasing function, we have an upper bound $UB(i,j)$ and a lower bound $LB(i,j)$ of the choice probability that i redeems j when j is offered to i:

$$UB(i,j) = \frac{\left(z_i^{\max(j)}\right)^{\lambda}}{1 + \left(z_i^{\max(j)}\right)^{\lambda}} \cdot \frac{e^{V(i,j)/\lambda}}{z_i^{\min(j)}} = \frac{\left(z_i^{\max(j)}\right)^{\lambda}}{1 + \left(z_i^{\max(j)}\right)^{\lambda}} \tag{27}$$

$$LB(i,j) = \frac{\left(z_i^{\min(j)}\right)^{\lambda}}{1 + \left(z_i^{\min(j)}\right)^{\lambda}} \cdot \frac{e^{V(i,j)/\lambda}}{z_i^{\max(j)}} = \frac{e^{V(i,j)}}{1 + e^{V(i,j)}} \cdot \frac{e^{V(i,j)/\lambda}}{z_i^{\max(j)}} \tag{28}$$

where $z_i^{\max(j)}$ and $z_i^{\min(j)}$ are respectively defined by (25) and (26):

Proposition 1. *Assume that $j \in \mathcal{J}$ is offered to $i \in \mathcal{I}$. The choice probability of i redeeming j is bounded by $UB(i,j)$ and $LB(i,j)$ defined by (27) and (28); i.e.,*

$$LB(i,j) \leq P(i,j) \leq UB(i,j).$$

Proof. Let $Y_i^{\mathrm{STB}(j)} \in \mathcal{Y}_i^{\mathrm{STB}(j)} \subset \mathcal{Y}_i^{\mathrm{STB}}$ indicate an independent set $\hat{V}_i^{\mathrm{STB}(j)}$ of $\hat{G}_i$ with (i,j); *i.e.*, $(i,j) \in \hat{V}_i^{\mathrm{STB}(j)}$ and $Y_i^{\mathrm{STB}(j)}(i,j) = 1$. Because $\hat{w}(z)$ is an increasing function, the choice probability of i redeeming j are bounded by $UB(i,j)$ and $LB(i,j)$ as follows:

$$P(i,j) = \hat{w}\left(z\left(Y_i^{\mathrm{STB}(j)}\right)\right) \frac{e^{V(i,j)/\lambda}}{z\left(Y_i^{\mathrm{STB}(j)}\right)} \leq \hat{w}\left(z_i^{\max(j)}\right) \frac{e^{V(i,j)/\lambda}}{z_i^{\min(j)}} = \hat{w}\left(z_i^{\max(j)}\right) = UB(i,j)$$

$$P(i,j) = \hat{w}\left(z\left(Y_i^{\mathrm{STB}(j)}\right)\right) \frac{e^{V(i,j)/\lambda}}{z\left(Y_i^{\mathrm{STB}(j)}\right)} \geq \hat{w}\left(z_i^{\min(j)}\right) \frac{e^{V(i,j)/\lambda}}{z_i^{\max(j)}} = LB(i,j)$$

Table 1 Important Cutting Planes for Enhanced($\lambda = 0.5$)

`Cutting Planes:`
`Gomory`
`Lift-and-project`
`Implied bound`
`MIR`
`Flow cover`
`RLT`
`Relax-and-lift`
`BQP`

Note: They were most frequently added to Enhanced(λ) with $\lambda = 0.5$ on the smallest network of ID = 9 with 52 potential customers in the default setting.

where

$$z\left(Y_i^{\mathrm{STB}(j)}\right) = \sum_{k \in \mathcal{J}} e^{V(i,k)/\lambda} Y_i^{\mathrm{STB}(j)}(i,k).$$

■

Constraint (8) in the MILP (24) can be strengthened by adding the following constraints:

$$LB(i,j)Y(i,j) \leq P(i,j) \leq UB(i,j)Y(i,j) \text{ for } i \in V \text{ and } j \in \mathcal{J}, \tag{29}$$

where $UB(i,j)$ and $LB(i,j)$ are respectively defined by (27) and (28). The MILP (24) strengthened by the bounding constraints (29) is denoted by Enhanced(λ).

Adding only important cutting planes also speeds up Enhanced(λ) significantly. Table 1 lists the important cutting planes found by preliminary experiments on small instances. The proposed enhancements introduced in this section will be verified by computational experiments in Section 4.

3.2. Heuristic

In this section, we introduce a heuristic algorithm. The heuristic algorithm first identifies an initial feasible solution and improves the initial solution by a local search algorithm. The local search algorithm repeats two steps: deletion and addition.

3.2.1. Initial Solutions. We consider two types of initial solutions: single and double options.

Single Options: The same option does not cause any conflict between friends, and any single option can be offered to all potential customers. The expected total revenue from offering a single option $j \in \mathcal{J}$ to all potential customers is

$$\sum_{i \in V} r(j)P(i,j) = \sum_{i \in V} \frac{r(j)e^{V(i,j)}}{1 + e^{V(i,j)}}$$

Double Options: If double options $j,k \in \mathcal{J}$ do not conflict to each other for any potential customers (*i.e.*, $\{(i,j),(h,k)\} \notin \hat{E}$ for all customer nodes $i,h \in V$), then the double options can be offered to all potential customers. The expected total revenue from offering such double options without any conflict is

$$\sum_{i \in V} \{r(j)P(i,j) + r(k)P(i,k)\}$$

where

$$P(i,j) = \frac{\left(e^{V(i,j)/\lambda} + e^{V(i,k)/\lambda}\right)^{\lambda}}{1 + \left(e^{V(i,j)/\lambda} + e^{V(i,k)/\lambda}\right)^{\lambda}} \cdot \frac{e^{V(i,j)/\lambda}}{\left(e^{V(i,j)/\lambda} + e^{V(i,k)/\lambda}\right)},$$

$$P(i,k) = \frac{\left(e^{V(i,j)/\lambda} + e^{V(i,k)/\lambda}\right)^{\lambda}}{1 + \left(e^{V(i,j)/\lambda} + e^{V(i,k)/\lambda}\right)^{\lambda}} \cdot \frac{e^{V(i,k)/\lambda}}{\left(e^{V(i,j)/\lambda} + e^{V(i,k)/\lambda}\right)}.$$

For example, Figure 1 illustrates $j = \$100$ and $k = \$300$ do not conflict with each other for Smart Phones 1 and 2.

If double options $j,k \in \mathcal{J}$ do conflict to each other for every potential customer (*i.e.*, $\{(i,j),(i,k)\} \in \hat{E}$ for all customer nodes $i \in V$), then the expected total revenue is the weight of the solution to the maximum weight independent set problem on the subgraph $\hat{G}[V \times \{j,k\}]$ induced by $V \times \{j,k\} \subset \hat{V}$ associating node weight

$$r(j)P(i,j) = \frac{r(j)e^{V(i,j)}}{1 + e^{V(i,j)}} \text{ on } (i,j) \in V \times \{j\} \subset \hat{V},$$

$$r(k)P(i,k) = \frac{r(k)e^{V(i,k)}}{1 + e^{V(i,k)}} \text{ on } (i,k) \in V \times \{k\} \subset \hat{V}.$$

Note that the maximum weight independent set problem is polynomially solvable for the bipartite subgraph $\hat{G}[V \times \{j,k\}]$ induced by two partites $V \times \{j\}$ and $V \times \{k\}$ within which there are no conflict graph edges. For example, Figure 1 illustrates $j = \$100$ and $k = \$200$ conflict to each other.

3.2.2. Local Search. Given an initial solution, the local search algorithm scans the conflict graph nodes, performing deletion and addition of the nodes. It repeats scanning until neither of deletion or addition improves the expected total revenue from any conflict graph nodes.

Deletion: Given a conflict graph node (u,q) in the existing solution (independent set), the deletion process (*i.e.*, $Y(u,q) = 1 \to Y(u,q) = 0$) deletes the node (u,q) from the solution if deleting it improves the expected total revenue.

Addition: For feasibility of the solution, adding a conflict graph node requires deleting its neighbors from the existing solution. If this addition of a conflict graph node (along with deleting its neighbors from the solution) improves the expected total revenue, the addition process adds it and deletes its neighbors from the existing solution.

Algorithm 1 describes the entire procedure of local search of repeating deletions and additions.

Algorithm 1 The local search method (LS)

Input: An initial feasible solution $Y^{\text{INITIAL}} \in \mathcal{Y}^{\text{STB}}$

Output: A local optimal solution Y^{FINAL} with $\text{REVENUE}^{\lambda}\left(Y^{\text{INITIAL}}\right) \leq \text{REVENUE}^{\lambda}\left(Y^{\text{FINAL}}\right)$

(Initialization) Set $Y = Y^{\text{INITIAL}}$ and $\mathsf{improved} = \mathsf{True}$
while $\mathsf{improved} = \mathsf{True}$ **do**
  $\mathsf{improved} = \mathsf{False}$
  for all conflict graph nodes $(i,j) \in \hat{V}$ **do**
    if $Y(i,j) = 1$ **then**
      $Y_i^{\text{Temp}} \leftarrow Y_i := (Y(i,k) : k \in \mathcal{J})$; $Y^{\text{Temp}}(i,j) \leftarrow 0 = 1 - Y(i,j)$
      if $\text{REVENUE}^{\lambda}(Y_i) < \text{REVENUE}^{\lambda}\left(Y_i^{\text{Temp}}\right)$ **then**
        $\mathsf{improved} = \mathsf{True}$
        $Y(i,j) \leftarrow 0$
        $\text{REVENUE}^{\lambda}(Y) \leftarrow \text{REVENUE}^{\lambda}(Y) - \text{REVENUE}^{\lambda}(Y_i) + \text{REVENUE}^{\lambda}\left(Y_i^{\text{Temp}}\right)$
      end if
    end if
    if $Y(i,j) = 0$ **then**
      $H := \{h \in V : (h,k) \sim (i,j) \in \hat{E}\} \cup \{i\}$
      $Y_H^{\text{Temp}} \leftarrow Y_H = (Y_h : h \in H)$; $Y_H^{\text{Temp}}(i,j) \leftarrow 1$; $Y_H^{\text{Temp}}(h,k) \leftarrow 0$ for $(h,k) \sim (i,j) \in \hat{E}$
      if $\text{REVENUE}^{\lambda}(Y_H) < \text{REVENUE}^{\lambda}\left(Y_H^{\text{Temp}}\right)$ **then**
        $\mathsf{improved} = \mathsf{True}$
        $Y_H \leftarrow Y_H^{\text{Temp}}$
        $\text{REVENUE}^{\lambda}(Y) \leftarrow \text{REVENUE}^{\lambda}(Y) - \text{REVENUE}^{\lambda}(Y_H) + \text{REVENUE}^{\lambda}\left(Y_H^{\text{Temp}}\right)$
      end if
    end if
  end for
end while
return $Y^{\text{FINAL}} = Y$

4. Numerical Experiments

We perform numerical experiments to verify the performance of our exact solution method for Enhanced(λ) and the proposed heuristic algorithm to tackle large-scale social networks. In Sections 4.1 and 4.2, we test the two algorithms on medium-scale instances of up to one thousand potential customers. Then, the heuristic algorithm tackles large-scale social networks of up to one million potential customers in Section 4.3.

Table 2 **Medium-Scale Instances**

ID	$\lvert V \rvert$	$\lvert E \rvert$	instances
9	52	146	50
5	61	270	50
3	150	1693	50
4	168	1656	50
2	224	3192	50
0	333	2519	10
8	534	4813	10
7	747	30025	10
6	786	14024	10
1	1034	26749	10

Note. Sorted in the increasing order of $\lvert V \rvert$.

Table 2 lists the ten Facebook Ego networks with varying size published by Leskovec and Mcauley (2012). An ego-network consists of the user nodes immediately connected to a user, called ego. The ego node is not included in the ego-network, but rather the ego-network consists only of his friends, who are referred to as alters. Throughout this section, we consider an electronics retailer planning a discount promotion campaign for three popular smartphones as described in Figure 1. His potential target customers are those who liked his store page on Facebook (*i.e.*, alters). For the ego-networks of less than 300 customers, 50 instances (inst0 - inst49) were generated. For the larger ego-networks, 10 instances were generated.

The computational experiments were performed on the Pittsburgh Supercomputer (Westropp 1996) running on 128 cores with 256 GB of RAM. The supercomputer optimally solved the MILPs running GUROBI 12.0.1 (Gurobi Optimization 2025). We use NetworkX (Hagberg et al. 2008) to manipulate graphs. To run the exact solution method and the heuristic algorithms, we wrote codes in Python 3 (Rossum and Drake 2009). All data, codes, and results are posted on GitHub.[1]

4.1. Exact Solution Method on Medium-Scale Instances

In this section, we optimally solve MILP-NL(λ) in (24) on medium-scale instances. In Section 4.1.1, we compare the default setting of the formulation with the enhanced formulation Enhanced(λ), and see that Enhanced(λ) solves the problem much faster than the default setting of the basic formulation (24). Then, Section 4.1.2 performs sensitivity analysis of the optimal solutions for log-sum parameters $\lambda = 0.5, 0.75, 1.00$. The analysis tells that the optimal solution Y_λ^{OPT} for $\lambda = 0.75$ is

[1] `https://github.com/s-shim/IAP_NL`

Table 3 **Default vs. Enhanced Formulations on** $\lambda = 0.5$

ID	$\|V\|$	$\|E\|$	instances	Default		Enhanced		Default / Enhanced		Row Type
				B&B	Time (s)	B&B	Time (s)	B&B	Time	
9	52	146	50	430,862	111.08	4,749	5.87	84.47	**18.17**	Mean
				96,574	17.16	578	0.47	16.54	2.15	SE
				3,648,444	747.72	19,999	17.06	570.16	110.89	max
				7,022	30.58	149	2.63	5.39	4.92	min
5	61	270	50	188,368	90.10	3,437	7.02	62.43	**12.34**	Mean
				52,307	9.64	600	0.46	12.73	0.54	SE
				2,362,063	465.57	26,791	22.30	544.13	27.66	max
				6,758	43.77	167	3.74	3.85	6.99	min
3	150	1,693	(3)	9,570,575	10,404.47	9,887	184.38	1,614.08	**62.74**	Mean
				1,110,220	391.62	1,088	5.33	312.35	12.25	SE
				11,123,053	11,158.79	40,133	273.58	2,142.23	85.47	max
				7,419,517	9,844.70	3,651	115.61	1,061.06	43.47	min
4	168	1,656	(2)	21,899,387	34,399.36	46,202	271.10	4,192.27	**196.99**	Mean
				15,366,612	13,343.01	14,438	13.43	2,956.40	115.28	SE
				37,265,999	47,742.37	490,047	523.24	7,148.67	312.28	max
				6,532,775	21,056.34	3,874	65.19	1,235.86	81.71	min

Note: Networks are listed in the increasing order of $|V|$. For the networks of ID = 3, three instances (inst0 - inst2) were sequentially solved. Since (Default) took longer than 24 hours for inst3, the next instances (inst3 - inst49) were not solved. Likewise, only two instances (inst0 and inst1) were solved on the network of ID = 4. Then none of the larger networks (ID = 2, 0, 8, 6, 7, 1) have been solved since (Default) took longer than 24 hours for inst0 of the larger networks.

near optimal for the problem in the other extreme cases $\lambda = 0.5, 1.00$. The analysis suggests to solve the problem of $\lambda = 0.75$ when the log-sum parameter λ is not available. Section 4.1.3 further speeds up Enhanced($\lambda = 0.75$) by warm starting with the MNL solution $Y^{\text{OPT}}_{\lambda'=1}$ to MILP-MNL (6)-(11) as the MILP-MNL can be solved very fast.

4.1.1. Performance of Enhancements. We perform computational experiments to verify the performance of the enhancements of MILP-NL(λ) in (24), which were introduced in Section 3.1. For log-sum parameter $\lambda = 0.5$, Table 3 presents the result from the computational experiments comparing the default setting of the MILP-NL(λ), denoted by (Default), with the enhanced setting of Enhanced(λ), denoted by (Enhanced). The (Enhanced) optimally solved the problem at least ten times faster than (Default) on ID = 9 and ID = 5 of 52 and 61 potential customers, respectively. On the larger instances (ID = 3 and 4) of more than a hundred customers, (Enhanced) performed 60 times to 200 times faster than (Default). Over two hundred customers, (Default) took more than 24 hours. In the remainder of this paper, (Enhanced) optimally solves MILP-NL(λ) when $\lambda < 1$.

4.1.2. Sensitivity Analysis. In this section, we perform a sensitivity analysis for the solutions to MILP-NL(λ) across three values of the log-sum parameter, $\lambda = 1.00, 0.75, 0.50$, and see that $\lambda = 0.75$ is a good estimate when $\lambda = \sqrt{1-\rho}$ is not available with the conceptual correlation

Table 4 Optimality Gap (%) on average across instances

ID	$\|V\|$	$\|E\|$	instances	$\text{REVENUE}^{\lambda}\left(Y_{\lambda'}^{\text{OPT}}\right)$			
				REVENUE^{λ}	$Y_{\lambda'=1.00}^{\text{OPT}}$	$Y_{\lambda'=0.75}^{\text{OPT}}$	$Y_{\lambda'=0.50}^{\text{OPT}}$
9	52	146	50	$\lambda = 1.00$	0.00%	**0.76%**	5.90%
				$\lambda = 0.75$	0.42%	**0.00%**	2.17%
				$\lambda = 0.50$	3.44%	**1.77%**	0.00%
5	61	270	50	$\lambda = 1.00$	0.00%	**0.79%**	5.05%
				$\lambda = 0.75$	0.47%	**0.00%**	1.67%
				$\lambda = 0.50$	3.61%	**1.79%**	0.00%
3	150	1693	50	$\lambda = 1.00$	0.00%	**0.62%**	5.83%
				$\lambda = 0.75$	0.33%	**0.00%**	2.28%
				$\lambda = 0.50$	3.11%	**1.84%**	0.00%
4	168	1656	50	$\lambda = 1.00$	0.00%	**0.64%**	5.72%
				$\lambda = 0.75$	0.36%	**0.00%**	2.26%
				$\lambda = 0.50$	3.03%	**1.67%**	0.00%
2	224	3192	50	$\lambda = 1.00$	0.00%	**0.67%**	5.65%
				$\lambda = 0.75$	0.37%	**0.00%**	2.22%
				$\lambda = 0.50$	2.98%	**1.56%**	0.00%
0	333	2519	(8)	$\lambda = 1.00$	0.00%	**0.69%**	5.46%
				$\lambda = 0.75$	0.40%	**0.00%**	1.97%
				$\lambda = 0.50$	3.51%	**2.02%**	0.00%
7	747	30025	(1)	$\lambda = 1.00$	0.00%	**0.62%**	6.03%
				$\lambda = 0.75$	0.34%	**0.00%**	2.60%
				$\lambda = 0.50$	3.06%	**1.77%**	0.00%

Note: Enhanced($\lambda = 0.50$) was optimally solved on the first eight instances (inst0 - inst7) of the network of ID = 0. On the network of ID = 7, the first instance was optimally solved, but the second instance (inst1) was not solved for 24 hours. The first instances (inst0) of ID = 8, ID = 6 and ID = 1 exceed the time limit of 24 hours.

coefficient ρ. Even though we are generalizing the MNL model (*i.e.*, $\lambda = 1.00$) into a NL model with $0 < \lambda \leq 1$, we reasonably suspect that λ is not very far from $\lambda = 1$ (or $\rho = 0$). In the remainder of this paper, we assume that $0.50 \leq \lambda \leq 1.00$ (*i.e.*, $0 \leq \rho \leq 0.75$).

We optimally solved MILP-NL(λ') and denote the optimal solution by $Y_{\lambda'}^{\text{OPT}}$. Assuming the true log-sum parameter λ, the sensitivity analysis evaluates the expected total revenue from the offer $Y_{\lambda'}^{\text{OPT}}$. Table 4 reports the optimality gaps:

$$\frac{\text{REVENUE}^{\lambda}\left(Y_{\lambda}^{\text{OPT}}\right) - \text{REVENUE}^{\lambda}\left(Y_{\lambda'}^{\text{OPT}}\right)}{\text{REVENUE}^{\lambda}\left(Y_{\lambda}^{\text{OPT}}\right)} \times 100\%$$

where $\text{REVENUE}^{\lambda}(Y)$ denotes expected total revenue (17) from a feasible offer Y with log-sum parameter λ. The optimal solution $Y_{\lambda'}^{\text{OPT}}$ to $\lambda' = 0.75$ is near-optimal for MILP-NL(λ) in the two extreme cases of $\lambda = 0.50$ and $\lambda = 1.00$ with at most 2% of small optimality gap. For more than 300 potential customers, Enhanced(λ) did not solve all ten instances within 24 hours. In the next section, we boost it by starting with an initial feasible solution, called warm-starting.

Table 5 **Enhanced ($\lambda = 0.75$) warm starting with MNL solutions**

ID	$\lvert V\rvert$	$\lvert E\rvert$	instances	MNL		Enhanced ($\lambda = 0.75$)		MNL / Enhanced		Row Type
				B&B	Time (s)	B&B	Time (s)	B&B	Time	
9	52	146	50	2	0.32	1861	3.67	1.95×10^{-3}	$\mathbf{9.96 \times 10^{-2}}$	Mean
				1	0.01	501	0.36	2.11×10^{-4}	3.82×10^{-3}	SE
				30	0.63	21965	14.31	6.10×10^{-3}	1.64×10^{-1}	max
				1	0.20	164	2.32	1.13×10^{-4}	2.75×10^{-2}	min
5	61	270	50	1	0.36	1926	4.97	2.16×10^{-2}	$\mathbf{7.55 \times 10^{-2}}$	Mean
				0	0.01	363	0.15	2.16×10^{-2}	7.55×10^{-2}	SE
				1	0.56	13099	7.45	1.00×10^{-0}	1.12×10^{-1}	max
				1	0.28	1	2.89	7.63×10^{-5}	4.94×10^{-2}	min
3	150	1693	50	2	4.19	7375	165.00	2.70×10^{-4}	$\mathbf{3.12 \times 10^{-2}}$	Mean
				0	0.12	345	8.65	5.53×10^{-5}	2.59×10^{-3}	SE
				17	6.27	17375	292.13	2.27×10^{-3}	9.29×10^{-2}	max
				1	3.18	1770	38.28	5.76×10^{-5}	1.47×10^{-2}	min
4	168	1656	50	2	5.32	7130	191.58	3.92×10^{-4}	$\mathbf{3.19 \times 10^{-2}}$	Mean
				0	0.16	280	8.59	1.22×10^{-4}	2.43×10^{-3}	SE
				15	9.11	11173	315.54	4.45×10^{-3}	9.65×10^{-2}	max
				1	3.77	2046	46.70	9.88×10^{-5}	1.55×10^{-2}	min
2	224	3192	50	3	11.53	7623	430.89	4.28×10^{-4}	$\mathbf{2.76 \times 10^{-2}}$	Mean
				1	0.30	199	14.25	2.51×10^{-4}	7.86×10^{-4}	SE
				57	18.24	10477	731.08	1.25×10^{-2}	4.12×10^{-2}	max
				1	8.22	3750	296.88	9.544×10^{-5}	1.57×10^{-2}	min
0	333	2519	10	33	13.08	25307	889.14	1.04×10^{-3}	$\mathbf{1.49 \times 10^{-2}}$	Mean
				30	0.86	8325	57.10	9.08×10^{-4}	6.88×10^{-4}	SE
				306	18.78	94343	1145.21	9.20×10^{-3}	1.87×10^{-2}	max
				1	9.83	6578	666.41	1.06×10^{-5}	1.12×10^{-2}	min
8	534	4813	10	17	33.59	81833	7279.58	1.76×10^{-4}	$\mathbf{6.76 \times 10^{-3}}$	Mean
				11	1.83	33728	1881.54	6.79×10^{-5}	1.03×10^{-3}	SE
				111	42.59	349925	21473.47	6.68×10^{-4}	1.07×10^{-2}	max
				1	26.11	7974	2665.94	2.86×10^{-6}	1.38×10^{-3}	min
7	747	30025	10	31	461.68	115188	46730.11	8.41×10^{-4}	$\mathbf{1.10 \times 10^{-2}}$	Mean
				5	8.97	56215	6081.25	3.14×10^{-4}	1.02×10^{-3}	SE
				60	500.24	595381	93810.52	3.33×10^{-3}	1.60×10^{-2}	max
				1	423.00	10812	31325.40	3.79×10^{-5}	4.57×10^{-3}	min
6	786	14024	10	40	200.09	59193	27159.64	1.14×10^{-3}	$\mathbf{8.30 \times 10^{-3}}$	Mean
				14	7.87	20643	3094.48	4.07×10^{-4}	1.03×10^{-3}	SE
				125	238.42	234353	45147.22	4.09×10^{-3}	1.51×10^{-2}	max
				1	158.06	14076	14452.31	4.27×10^{-6}	4.81×10^{-3}	min

Note: Enhanced($\lambda = 0.75$) were optimally solved on none of the 10 instances of Network ID = 1 within 24 hours. For the result on Network ID = 1, see Table 6.

4.1.3. Enhanced($\lambda = 0.75$) Warm Starting with MNL Solutions. In this section, we optimally solve Enhanced($\lambda = 0.75$) on the medium-scale instances, starting with the MNL solutions (*i.e.*, for $\lambda' = 1$) that can be solved quickly by MILP-MNL (6)-(11). The optimal solution $Y^{\mathrm{OPT}}_{\lambda'=1}$ to MILP-MNL can warm start Enhanced($\lambda = 0.75$). That is, we can run a MILP solver

Table 6 **Enhanced ($\lambda = 0.75$) warm starting with the MNL solutions**

ID	$\|V\|$	$\|E\|$	Time Limit (hours)	Instance	Enhanced ($\lambda = 0.75$)		
					B&B	Improve	Gap
1	1034	26749	24	inst0	19481	0.38%	0.60%
				inst1	14213	0.36%	0.59%
				inst2	13604	0.35%	0.79%
				inst3	22336	0.37%	0.13%
				inst4	15733	0.35%	0.30%
				inst5	15047	0.35%	0.42%
				inst6	12277	0.35%	0.57%
				inst7	11418	0.37%	0.67%
				inst8	14361	0.32%	0.48%
				inst9	13143	0.34%	0.56%

Note: Improve (%) $= \frac{\text{Final}-\text{MNL}}{\text{MNL}} \times 100\%$ and Gap (%) $= \frac{\text{bestBd}-\text{Final}}{\text{Final}} \times 100\%$

(GUROBI) with the initial feasible solution $Y^{\text{OPT}}_{\lambda'=1}$. Table 5 presents the result from the mixed-integer linear programming. The initial time to solve MILP-MNL is less than 10% of the time to solve Enhanced($\lambda = 0.75$). This warm start significantly speeds up the computational time of mixed-integer linear programming.

Table 5 does not report the largest network (ID = 1) because none of the 10 instances were solved within 24 hours, even with warm-starting with the MNL solutions. The final solutions to the 10 instances are reported in Table 6, in which the MNL solutions (MNL) are reported to be improved to the final solutions (Final) by

$$\text{Improve (\%)} = \frac{\text{Final} - \text{MNL}}{\text{MNL}} \times 100\%$$

The gap between the final integer solution (Final) and the best bound (bestBd) given by the fractional optimal solution at the end of the time limit (24 hours) is reported as

$$\text{Gap (\%)} = \frac{\text{bestBd} - \text{Final}}{\text{Final}} \times 100\%$$

The gaps (Gap) are all less than 1%, and the final solutions are guaranteed to be near optimal.

4.2. Performance of Heuristic on Medium-Scale Instances

In this section, the heuristic algorithm introduced in Section 3.2 tackles medium-scale instances (Table 2), which are optimally solved in Section 4.1.3 by Enhanced($\lambda = 0.75$) with warm-starting from the MNL solutions. We compare the optimal solutions with the heuristic solutions and verify the quality of the heuristic solutions. We also compare the computational time of the heuristic algorithm with that of Enhanced(λ).

The proposed heuristic algorithm, a local search (LS) method, performs best with parallel processing, starting with multiple initial solutions. For our example of six options, six cores simultaneously perform the six iterations of the LS algorithm, denoted by Single-LS (SLS), beginning with

Table 7 **Number of Options offered by Single-LS (SLS) and Optimality and Time Ratio of SLS vs. Enhanced ($\lambda = 0.75$)**

ID	$\|V\|$	$\|E\|$	instances	Offered by SLS		SLS / Enhanced Solutions			Row Type
				Initial	Final	Initial	Final	Time	
9	52	146	50	52.0	125.7	54.65%	99.54%	9.10%	Mean
				0.0	0.7	3.46%	0.09%	0.48%	SE
				52.0	135.0	83.86%	100.00%	18.09%	max
				52.0	112.0	20.01%	96.79%	1.55%	min
5	61	270	50	61.0	148.2	54.33%	99.49%	10.54%	Mean
				0.0	0.7	3.35%	0.09%	0.45%	SE
				61.0	159.0	82.36%	100.00%	18.88%	max
				61.0	135.0	22.03%	96.14%	4.38%	min
3	150	1693	50	150.0	369.0	49.49%	99.63%	5.19%	Mean
				0.0	1.4	3.48%	0.09%	0.58%	SE
				150.0	395.0	81.39%	100.00%	22.99%	max
				150.0	337.0	21.39%	97.26%	1.60%	min
4	168	1656	50	168.0	411.7	48.04%	99.63%	4.91%	Mean
				0.0	1.8	3.37%	0.09%	0.50%	SE
				168.0	434.0	81.98%	100.00%	17.99%	max
				168.0	367.0	21.93%	96.64%	2.12%	min
2	224	3192	50	224.0	551.1	44.66%	99.63%	4.58%	Mean
				0.0	2.2	3.34%	0.10%	0.19%	SE
				224.0	570.0	81.86%	100.00%	7.21%	max
				224.0	462.0	22.12%	95.16%	2.02%	min
0	333	2519	10	333.0	802.3	56.02%	99.23%	3.01%	Mean
				0.0	6.2	7.86%	0.14%	0.26%	SE
				333.0	823.0	80.60%	99.89%	4.42%	max
				333.0	762.0	22.48%	98.49%	1.76%	min
8	534	4813	10	534.0	1311.2	31.31%	99.01%	2.46%	Mean
				0.0	2.9	4.58%	0.16%	0.47%	SE
				534.0	1325.0	59.11%	99.82%	4.68%	max
				534.0	1297.0	22.58%	98.13%	0.45%	min
7	747	30025	10	747.0	1828.3	48.90%	99.42%	1.45%	Mean
				0.0	16.9	7.11%	0.19%	0.17%	SE
				747.0	1873.0	80.41%	100.00%	2.17%	max
				747.0	1687.0	22.38%	97.88%	0.67%	min
6	786	14024	10	786.0	1938.2	33.87%	99.54%	1.72%	Mean
				0.0	2.5	5.27%	0.18%	0.20%	SE
				786.0	1954.0	58.45%	99.97%	2.74%	max
				786.0	1927.0	22.56%	97.97%	0.87%	min
1	1034	26749	10*	1034.0	2540.3	39.11%	99.56%	1.10%	Mean
				0.0	7.7	6.88%	0.17%	0.05%	SE
				1034.0	2573.0	80.58%	99.99%	1.32%	max
				1034.0	2500.0	22.14%	98.37%	0.72%	min

Note: * Enhanced($\lambda = 0.75$) were optimally solved on none of the 10 instances of Network ID = 1 within 24 hours. The solutions to Heuristic were compared against the best solutions (to Enhanced($\lambda = 0.75$)) that are analyzed in Table 6.

the six single options. Table 7 presents the iterations that produce the best final solutions, each of which is from among the six iterations for an instance. The number of offered single options is initially the same as the total number of potential customers in the social network. The optimality of the final solutions (*i.e.*, ratio of the expected total revenues of the final solution to the optimal solution) are all over 99% on average, while the optimality of the initial solutions are less than 60%. The computational time of the local search algorithm is up to 10% of the exact solution time by Enhanced($\lambda = 0.75$). The computational time becomes relatively better and better as the number of customers grows.

Beginning with fifteen pairs of double options ($i.e$, $\binom{\mathcal{J}}{2}$), Table 8 reports the result from the fifteen simultaneous iterations of the local search algorithm, denoted by Double-LS (DLS). The optimality of the initial solution of DLS is much higher than that of SLS. While the initial solutions take some time on three iterations to solve the maximum independent set problems on the bipartite graphs capturing three conflicting double options (*i.e.*, $\{\$100, \$200\}, \{\$300, \$400\}, \{\$500, \$600\}$ for Smart Phones 1, 2 and 3), the computational time of DLS is shorter than that of SLS to reach the final solution, starting with the initial solutions that are close to optimality. Across the medium-scale networks, the final solutions of DLS are uniformly better than those of SLS on average. On none of the 10 instances of Network ID = 1, Enhanced($\lambda = 0.75$) were optimally solved within 24 hours. The final solutions (Final) to the heuristic algorithm were compared against the best solutions (to Enhanced($\lambda = 0.75$)) that are analyzed in Table 6.

4.3. Large-Scale Social Networks of 200,000 and 1,000,000 Potential Customers

In this section, the proposed heuristic algorithms tackle two large-scale networks of 200,000 and 1,000,000 potential customers (Cho et al. 2011, Yang and Leskovec 2015), which we call the Gowalla and YouTube networks in this paper.

Gowalla is a location-based social networking website where users share their locations by checking-in. The friendship network is undirected and consists of 196,591 nodes and 950,327 edges. We generated random 10 instances on the Gowalla network. Table 9 reports the result from the computational experiments on the ten instances. Each instance is solved by 21 multiple simultaneous iterations, beginning with 6 single options and 15 double options. The multiprocessing of our heuristic are done within an hour.

Youtube is a video-sharing web site that includes a social network. In the Youtube social network, users form friendship each other and users can create groups which other users can join. Cho et al. (2011) considered such user-defined groups as ground-truth communities. This data had been first provided by Mislove et al. (2007). Cho et al. (2011) regarded each connected component in a group as a separate ground-truth community. They provide the top 5,000 communities with the highest

Table 8 Number of Options offered by Double-LS (DLS) and Optimality and Time Ratio of DLS vs. Enhanced ($\lambda = 0.75$)

ID	$\|V\|$	$\|E\|$	instances	Offered by DLS		DLS / Enhanced Solutions			Row Type
				Initial	Final	Initial	Final	Time	
9	52	146	50	94.6	126.8	85.01%	99.89%	4.57%	Mean
				2.8	0.6	3.24%	0.01%	0.50%	SE
				104.0	135.0	99.42%	100.00%	17.27%	max
				52.0	112.0	25.99%	99.42%	0.58%	min
5	61	270	50	120.7	150.2	93.33%	99.77%	3.60%	Mean
				1.2	0.6	1.91%	0.03%	0.30%	SE
				122.0	161.0	99.22%	100.00%	15.63%	max
				61.0	137.0	27.87%	99.18%	1.37%	min
3	150	1693	50	294.0	372.6	96.30%	99.98%	1.48%	Mean
				4.1	0.8	1.18%	0.00%	0.23%	SE
				300.0	395.0	99.14%	100.00%	10.98%	max
				150.0	363.0	57.83%	99.68%	0.44%	min
4	168	1656	50	336.0	416.5	97.14%	99.98%	1.15%	Mean
				0.0	1.0	0.97%	0.00%	0.11%	SE
				336.0	434.0	99.20%	100.00%	3.98%	max
				336.0	395.0	64.13%	99.77%	0.50%	min
2	224	3192	50	425.6	555.4	91.13%	99.97%	1.34%	Mean
				9.6	1.1	3.13%	0.00%	0.15%	SE
				448.0	572.0	99.10%	100.00%	6.06%	max
				224.0	535.0	24.55%	99.82%	0.58%	min
0	333	2519	10	632.7	821.0	94.47%	99.90%	0.87%	Mean
				33.3	2.1	4.20%	0.02%	0.32%	SE
				666.0	830.0	98.98%	100.00%	3.80%	max
				333.0	812.0	56.63%	99.76%	0.40%	min
8	534	4813	10	1068.0	1322.3	98.73%	99.96%	0.30%	Mean
				0.0	2.2	0.03%	0.01%	0.05%	SE
				1068.0	1329.0	98.92%	100.00%	0.53%	max
				1068.0	1309.0	98.56%	99.91%	0.06%	min
7	747	30025	10	1494.0	1850.5	95.37%	99.99%	0.32%	Mean
				0.0	5.9	3.40%	0.00%	0.08%	SE
				1494.0	1873.0	98.94%	100.00%	1.03%	max
				1494.0	1816.0	64.76%	99.97%	0.11%	min
6	786	14024	10	1572.0	1945.4	98.74%	99.97%	0.23%	Mean
				0.0	2.6	0.03%	0.00%	0.02%	SE
				1572.0	1959.0	98.87%	99.99%	0.36%	max
				1572.0	1933.0	98.61%	99.92%	0.11%	min
1	1034	26749	10*	2068.0	2551.2	98.81%	99.98%	0.15%	Mean
				0.0	4.9	0.02%	0.00%	0.00%	SE
				2068.0	2574.0	98.94%	100.00%	0.17%	max
				2068.0	2530.0	98.69%	99.96%	0.14%	min

Note: * Enhanced($\lambda = 0.75$) were optimally solved on none of the 10 instances of Network ID = 1 within 24 hours. The final solutions (Final) were compared against the best solutions (to Enhanced($\lambda = 0.75$)) that are analyzed in Table 6.

Table 9 Revenue ($) from the options offered by two local search algorithms on a large-scale network

ID	Instance	Single-LS			Double-LS		
		Initial ($)	Final ($)	Time (s)	Initial ($)	Final ($)	Time (s)
Gowalla	0	16,598,579.79	**28,889,143.21**	1250.0	15,817,516.54	**28,889,035.92**	2034.3
	1	7,528,196.79	**28,879,468.28**	1708.1	7,528,196.79	**28,879,468.28**	2284.0
	2	7,540,066.39	**28,882,359.05**	1259.7	28,471,588.15	**28,884,416.69**	1971.9
	3	7,506,403.60	**28,883,009.30**	1363.3	28,483,223.95	**28,888,518.61**	2091.9
	4	16,634,082.65	**28,881,494.30**	1497.0	28,475,139.02	**28,882,826.41**	2222.8
	5	6,530,823.55	**28,867,461.18**	1218.8	28,498,141.24	**28,902,211.44**	2006.2
	6	16,614,825.42	**28,891,128.36**	1371.3	28,484,662.37	**28,892,383.88**	2071.9
	7	23,139,306.80	**28,831,345.11**	1382.6	28,496,210.77	**28,897,062.98**	2076.2
	8	6,535,531.88	**28,838,676.82**	1489.8	28,465,020.95	**28,874,299.58**	2218.0
	9	6,532,131.84	**28,853,557.21**	1506.5	28,449,219.30	**28,857,306.33**	2177.0

Note: Gowalla network has 196,591 customers and 950,327 friendship relations. Log-sum parameter is set to $\lambda = 0.75$.

Table 10 Revenue from the options offered by two local search algorithms on a very large-scale network

ID	Method	Initial				Final		
		Available	Offered	Revenue ($)	Time (s)	Offered	Revenue ($)	Time (s)
YT	SLS	1	1,134,890	37,733,344.97	101.5	2,737,875	167,726,029.20	94,162.5
		2	1,134,890	43,404,169.39	154.2	2,733,004	167,084,575.50	87,061.6
		3	1,134,890	95,960,297.99	222.0	2,744,287	167,587,553.80	91,716.5
		4	1,134,890	71,961,073.89	290.8	2,586,252	167,041,797.20	86,853.0
		5	1,134,890	133,508,623.60	347.6	2,702,706	**167,907,368.10**	79,595.7
		6	1,134,890	88,235,786.28	397.8	2,752,862	157,103,324.90	117,061.3
	DLS	1 & 2	1,134,890	43,404,169.39	10,721.6	2,733,004	167,084,575.50	77,812.1
		1 & 3	2,269,780	91,405,024.69	131.3	2,744,274	167,588,383.50	112,184.2
		1 & 4	2,269,780	73,972,249.64	159.1	2,580,072	166,467,190.90	123,201.0
		1 & 5	2,269,780	112,447,203.20	196.5	2,721,696	167,813,217.20	116,722.9
		1 & 6	2,269,780	80,984,802.56	259.8	2,752,517	157,184,399.00	151,198.7
		2 & 3	2,269,780	107,505,372.70	284.4	2,743,396	167,642,997.10	117,153.1
		2 & 4	2,269,780	89,678,966.25	331.4	2,587,037	167,239,077.00	108,759.0
		2 & 5	2,269,780	137,003,376.80	381.6	2,705,200	**167,909,960.80**	104,276.0
		2 & 6	2,269,780	100,876,071.70	418.2	2,745,693	157,182,794.40	145,274.6
		3 & 4	1,134,890	71,961,073.89	15,164.2	2,586,252	167,041,797.20	128,140.9
		3 & 5	2,269,780	164,389,861.80	555.6	2,748,783	167,672,816.90	100,767.8
		3 & 6	2,269,780	137,304,560.00	609.6	2,769,028	157,476,722.90	152,329.9
		4 & 5	2,269,780	160,479,440.20	650.9	2,599,019	166,902,797.30	133,257.5
		4 & 6	2,269,780	127,567,616.70	719.3	2,623,155	153,745,790.20	143,616.9
		5 & 6	1,134,890	88,235,786.28	13,368.9	2,752,862	157,103,324.90	110,346.0

Note: YouTube (YT) network has 1,134,890 customers and 2,987,624 friendship relations. Log-sum parameter is set to $\lambda = 0.75$.

quality. The YouTube network used in our paper is the largest connected component with 1,134,890 users. We generate one random instance. Table 10 reports the result from the computational experiment on the instance. For the YouTube network, our heuristic algorithm is done within two days.

5. Conclusion

We introduced a more realistic model of the inequity aversion pricing problem under a nested logit demand function and a mixed-integer linear program (MILP) of the nested logit model. We enhanced the MILP by bounding the continuous variables of choice probabilities and adding only important cutting planes. We further sped up the enhanced MILP by warm-starting with the multinomial (MNL) logit solution, and optimally solved the MILP on a thousand potential customers offering six discount price options.

To tackle large-scale social networks, we developed a heuristic algorithm that begins with an initial feasible offer and reaches a local optimum. The Pittsburgh Supercomputer performed parallel processing starting with single and double options. On a thousand potential customers, the computational experiments verified the heuristic algorithm to reach near-optima extremely fast. We successfully ran the multiprocessing of the heuristic algorithm on a million potential customers within two days.

This paper assumed two nests: a singleton nest of the base case (no-purchase) and the other nest of redeeming the offered discount price options. While the simple assumption allowed us to linearize the NL model of the inequity aversion pricing problem, the proposed heuristic algorithm can tackle not only large-scale social networks but also the most general cases of the NL model with more than two nests.

References

Alon N, Mansour Y, Tennenholtz M (2013) Differential pricing with inequity aversion in social networks. In *ACM Conference on Economics and Computation, EC 2013*, 9—24.

Amanatidis G, Fulla P, Markakis E, Sornat K (2021) Inequity Aversion Pricing over Social Networks: Approximation Algorithms and Hardness Results. *Theoretical Computer Science* **871**, 62–78.

Bolton GE, Ockenfels A (2000) A theory of equity, reciprocity and competition. *American Economic Review* **100**, 166—193.

Cho E, Myers SA, Leskovec J (2011) Friendship and mobility: user movement in location-based social networks. In *KDD 2011: Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, 1082–1090. `https://doi.org/10.1145/2020408.2020579`

Chopra S, Rao MR (1993) The partition problem. *Mathematical Programming* **59**, 87–115.

Chopra S, Park H, Shim S (2022) Extended graph formulation for the inequity aversion pricing problem. *INFORMS Journal on Computing*, Published online in Articles in Advance 02 Feb 2022, *URL https://doi.org/10.1287/ijoc.2021.1148*

Chopra S, Shim S (2020) A strong formulation for the graph partition problem. *Networks* **75**, 183–202.

Fehr E, Schmidt KM (1999) A theory of fairness, competition and co-operation. *Quarterly Journal of Economics* **114**, 817—868.

Gurobi Optimization, LLC. 2025. Gurobi Optimizer Reference Manual. https://www.gurobi.com

Hagberg A, Swart P, Chult DS (2008) Exploring network structure, dynamics, and function using networkx. Technical report, Los Alamos National Laboratory (LANL), Los Alamos, NM (United States).

Heiss, F. 2002. Structural choice analysis with nested logit models. *The Stata Journal* **2** (3) 227–252.

Leskovec J, Mcauley JJ (2012) Learning to discover social circles in ego networks. *Advances in Neural Information Processing Systems*, 539–547.

Ljubíc, I. and E. Moreno. 2018. Outer approximation and submodular cuts for maximum capture facility location problems with random utilities. *European Journal of Operational Research* **266** 46–56.

McFadden D (1974) Conditional logit analysis of qualitative choice behavior. In *Frontiers in Econometrics*, ed. P. Zerembka, 105–142. New York: Academic Press.

Mislove A, Marcon M, Gummadi KP, Druschel P, Bhattacharjee B. (2007) Measurement and analysis of online social networks. In *IMC 2007: Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*, 29–42. `https://doi.org/10.1145/1298306.1298311`

Rossum, G. and Drake, F.L., 2009. Python 3 Reference Manual, CreateSpace, Scotts Valley, CA, USA.

Westropp, J.C., 1996. IEEE Computational Science and Engineering. *IEEE Computational Science and Engineering* **3** (1) 8–12.

Yang, J. and J. Leskovec. 2015. Defining and evaluating network communities based on ground-truth. *Knowledge and Information Systems* **42** (1) 181–213.