Recommender systems with capacity

Final Project Report
Computational Social Processes CSCI-4110/6110

Seyed Shahab Mofidi
Anders Maraviglia
Fall 2016

1 Introduction

Let us start by a familiar example most readers can relate to. As a host, I prepared food for a group of friends whom I invited to my house. Since a few of my guests were vegetarian, I had to prepare two dishes; veggie option and non-veggie with some forecasts of how many of my guests are omnivore and how many are not. As usual, a margin of error was considered in my prediction and two dishes were prepared. On the night of the event, my omnivore guests were interested in the veggie dish more and ate the food they were not supposed to, let my vegetarian guests leave with empty plates. Many what if scenarios struck my mind after that event. But the main question was how to design a better mechanism that ensures a better outcome.

Through this example, we illustrate a central mechanism (the host) who recommends a set of available alternatives (dish(es)) to multiple agents (guests). Undoubtedly, this example is a simplified version of what organizations, firms and companies deal with as a decision making problem to understand how to design a recommendation system that guarantee systematic performance while their members, users, and customers preferences are also considered. In fact, the problem of our interest in this paper, applies to environments that share five fundamental characteristics which we highlight here. **(1)** ***Hierarchical decision making process***. The problem of our interest is similar to leader-follower game models in terms of the decision makers roles in which the leader (i.e., host in this case) should decide and move first (i.e., prepare dishes) with some perceptions of the follower(s) (i.e., agents) reactions. After the leader makes decisions, the agents decide and move afterwards (i.e., what to eat). Moreover, the leader has no control on the agents choice. (i.e., the host cannot force guests what to eat). **(2)** ***Limited available resources***. When infinite amount of a resource is available, recommendations by the system is only centered on the preferences of each agent isolatedly since even if all agents' first choice is alternative "a" for example, all can obtain their first choice. However, when a resource is relatively scarce to the number of agents interested, challenges arise as who should take priority among agents, and what are the associated costs to the system as the consequences. Study these challenges and consequences is to our interest. In the dinner party example, imagine if more food had been prepared, the outcome would not have been the same. However, it comes at a higher price of sacrificing host's budget for an ordinary dinner or a larger amount of food leftovers. **(3)** ***Simultaneous personalized recommendations***. Agents receive recommendations either all at the same time (simultaneous) or in an order (sequential) through a time horizon. In sequential recommendations, system recommends to each agent one at a time. The set system recommends to a new agent can be updated based on the choice(s) of the previous agent. Hence, the recommendation's context depends on the place of the agent in the queue. And so the problem boils down to what should the order of the queue be (imagine the host make a line at the dinner table and put vegetarian guests first) which is out of scope of this work. In simultaneous recommendations, system's decision-making process happens once and agents' choices come afterwards. The simultaneous recommendation set(s) to agents are often aggregate in the system. (i.e., single recommendation available for group(s) of agents. E.g., one food table for all guests at the dinner party). We are interested in the simultaneous *personalized* recommendations problem which is very complicated specifically when resources are finite. Because in this case, the system needs to incorporate all agents' preferences in advance and thus need more accurate information about the agents choice. In the above example, the problem could have been solved if the food is served to each guest individually, however, guests should have filled a questionnaire in advance to provide information on their diet, preferences, food allergies, etc. **(4)** ***Myopic independent rational agents***. When you are at the table as a guest, your choice of food only depends on your personal preference and almost nobody considers how his/her choice can affect others as it was the case in this example, and what the omnivore guests did. However, the overall outcome could be improved if the system intervenes with it's holistic view. Here, limiting the omnivores choice to the non-veggie option (i.e. not offering them to the vegetarian dish) would be a proper example of system's intervention. Therefore, when agents are myopic independent rational individuals, the personalized recommendation set can be used as an intervention tool for the system to guarantee the achievement of some major concerns. **(5)** ***Two objectives are not necessarily aligned***. In these types of problems, leader's objective is to maximize benefit (minimize costs) while agents are utility maximizers and sometimes these two objectives do not agree. Here in this example, omnivore guests who ate veggie dish maximized their own utility while the host's objective was to satisfy all of his guests.

Underlying many applications in recommendation systems are these five characteristics. One of the emerging application is the "*sharing economy*" with market valued at $75 billion [3]. The sharing economy is where resources owned by independent entities are collectively shared through a central mechanism with internet-based platforms who owns no resources; instead operates a marketplace (e.g., companies like Uber, Airbnb, TaskRabbit, etc.). Critical to a central mechanism's success is its ability (1) to entice large participation by agents and (2) to accommodate their preferences to ensure repeat participation. Agents expect high-quality service (e.g., tasks completed to desired specifications within a given time). This is achieved when the central mechanism takes a systematic view of allocating limited resources to agents. Agents benefit from a large pool of alternatives and when the central mechanism retains some control to ensure service expectations are met. Agents desire autonomy and discretion to decide when and how they want to provide access to their resources based on their individual preferences. If discretion is not provided, this limits agent participation. Thus, success in sharing economy depends on agents satisfaction. However, agents preferences are not always aligned with system objectives. Take for example a ride-sharing application, like Uber or Lyft. The agents are drivers that prefer riders with destinations on their current route or to a high populated area (where another rider is likely). The central mechanism wants to maximize the number of successful matches by assigning the closest driver to a rider.

Another application is in the retail industry where, a company like Macy's wants to offer it's customers a personalized set of in-store coupons of limited inventory through emails. These coupons are tailored based on limited leftover inventory of some retail locations or regional clearances. Therefore, there are couple of products here and there that Macy's wants to offer to specific customers at a lower price. Crucial to maximize revenue is Macy's ability to offer the right set(s) of coupons to the right group(s) of customers such that no customer receive excessive list of in-store coupons which are not to her interest (guarantee her return to the store) and are not available in near-by locations (guarantee her satisfaction). Again worth noting that the inventory is limited and the challenge is limiting some customers to receive all coupons since Macy's cannot guarantee to satisfy all demand. Moreover, customers also have the discretion not to buy any products and hence, at the same time incentives exist for Macy's to offer same coupons to a larger group of customers.

One more application is in the emerging electric car industry, mobile application companies (e.g., PlugShare) recommend charging station alternatives to electric car users. However, the number of charging stations are limited and the charging process is not as fast as pumping your car at a gas station yet and each car occupies stations for a long period of time. PlugShare should also take the congestion factor of stations as well as your convenience preference of the nearest and fastest service into consideration when making recommendations.

We also presented our research project through the course project assignment motivation where each group has to choose a project topic from a list of different topics available by the instructor. All projects are available to all groups in the class. No group can choose more than one topic and two groups can not work on the same project. Moreover, the instructor has preferences over the assignments (i.e., which topic is done by which group) due to particular skills needed for a specific project. For example, one project is a software development project where a special programming language is needed that only few groups are able to deploy. Difficulties arise when a topic is left unselected (i.e., rejected by all groups) and some topics are selected by two or more groups (i.e., duplicates occur). Therefore, the instructor has to intervene and adjust the assignments considering the ability of groups as well as their preferences.

In all fo these examples, all of the five mentioned characteristics are present. Therefore, this research is interested in determining the *optimal personalized assortment* to recommend to each agent in these environments. By optimality, we mean maximum systems benefit with regards to the decision behavior of rational utility maximizer agents. In deed, the agents' decisions create dependencies in the system that impact system benefit when limited resources are available. We describe two common approaches that currently exist for the central mechanism design of recommender systems which can be considered as two extreme cases for the spectrum which we study. **(1)** ***Centralized***, and **(2)** ***Decentralized*** approaches.

In the centralized approach, system is the only decision maker and the agents follow orders without any discretion. System performance is the only objective, the choices are dictated to the agents, and their preferences are not taken into account (i.e., either ignored by the system or their preferences are not available to the system). An example of this approach is the well-known Vehicle Routing Problem (VRP). In VRP, fleet dispatcher (central mechanism) matches the optimal set of destinations and corresponding routes to a fleet of vehicles in order to meet the demand of certain nodes with optimal systematic performance (e.g., minimum cost). In environments where agents are committed to the system and are employed by the company, the only objective is to improve the performance of the system, and the agents either do not have any preferences or their preferences are not taken into consideration in the solution space of the system. Moreover, the central decision making assures the solution to be optimal, no resource constraint is violated, double efforts and underutilization is minimized (i.e., two agents do not visit single node and empty travels are at the minimum possible). The decision making in the centralized approach often needs to be fast. In addition, achieving the optimal solution depends on the full compliance of the agents which is not a false assumption in this case. Another example would be the job assignment problem where tasks are optimally assigned to machines for different time spans. Obviously machines have neither preferences over the jobs nor the ability to disobey the assigned jobs.

On the other end of this spectrum is the decentralized approach where agents are the only decision-making individuals with personal preferences. All alternatives are available to all agents who make selections based on their own preferences, and the system does not limit agent's options. A decentralized approach applies in environments where social welfare (i.e., sum of all agents' utilities) of the agent's are at the top priorities (e.g., customer satisfaction). This is due to the fact that agents often have no commitments to the system and hence, no dictatorship applies here. The central mechanism usually facilitates agents' decision making process through personalized recommendations which are very well-known as recommender systems (e.g., Google, Netflix, etc.). If you are searching for a hotel to spend two nights of your vacation in, of course Google can not force you to pick a choice but can navigate you through a list of endless choices. These types of recommender systems employ machine learning techniques (i.e., collaborative and content-based filtering are two extremely popular methods and interested readers are referred to [50]) to presume agents preferences and increase agents participation to improve the chance of matching an agent with an alternative. In a decentralized approach, systematic performance is ignored, and high utilization is compromised due to multiple agents' common interests in some alternatives lead to other alternatives left unselected. Therefore, decentralized control can result in reduced systematic performance [51]. For example, decreasing the number of unoccupied room of a hotel, visitor's travel distances, city congestions are not a concern in Google recommender systems. Moreover, number of alternatives recommended are hypothetically infinite and decision making process for agents receives more effort compared to when the central mechanism makes decisions. Table 1) summarizes the advantages and disadvantages of these two approaches.

Neither of these two extreme approaches are applicable when the five mentioned characteristics exist in a recommendation system because neither are able to fully harness underutilized resources. For example, in a centralized approach, Uber's central mechanism takes a rider-centric view by recommending a single rider's alternative to the closest driver (ignoring driver's destination preferences). The driver has 15 seconds to either accept or reject the recommended alternative. Uber has policies to strongly encourage high driver acceptance rates. Uber's current centralized approach limits who can participate as a driver because drivers must be committed to Uber tasks. If drivers want to interleave a ride-share with their planned travels, they cannot because the central mechanism does not take into account driver preferences when recommending rides. In a centralized approach with limited discretion, agents preferences are not taken into account. Hence, it affects systematic performance negatively. Also, decentralize approach will not fit the Macy's example. All coupons cannot be offered to all customers. By the time they come to the store, the deal might not be available at that store, or some other customer might already have bought the specific product and either the customer leave the store dissatisfied or Macy's has to reduce the price of a substitute product which Macy's had not intended to do so. The reader can easily verify why a centralized is not applicable here. Also, for the charging station example, PlugShare company takes decentralized approach and shows the specification (e.g., locations, number of stations, charging adaptors, etc.) of all nearby stations to all drivers and lets them choose. Although it also shows customers the data on the number occupied slots, it does not have any control on drivers choice. For example if only one slot is available and two cars pick the same stations as their destination, one of them would be dissatisfied and has to either wait or pick another further station which adds to the inefficiency of the system. If the system recommended some other alternatives, higher utilization could be achieved. A smarter and more efficient way is to consider the user's preferences (e.g., the car type, convenient locations, etc.) as well as the overall systematic performance (e.g., demand of a particular station, congestion, other users preferences, etc.) when making recommendations.

Therefore, a new approach is needed to combine the advantages of a centralized approach (e.g., less participant effort, quicker time to match, systematic resource allocation) with the advantages of a decentralized approach (e.g., agent discretion and privacy, increased participation opportunities). We study a hybrid approach that provide agents with an assortment of limited personalized recommendations. This hybrid mechanism is a hierarchical approach in which the central mechanism considers agents' preferences, as well as systematic performance and interdependencies, when creating personalized recommendations. First, the central mechanism makes personalized recommendations consisting of a set of alternatives to multiple distributed independent agents. Then, agents have discretion to make selections or rejections based on their own unique valuation of the recommended set. Observed system outcomes are a function of what is recommended and decentralized selections. This approach enables holistic resource allocation and can enforce service level requirements. Also, agents retain autonomy and discretion. Personalized recommendations eliminates the need to evaluate a large numbers of alternatives, reduce the effort required from participants, and thus, can increase agents satisfaction. Our model facilitates this type of central mechanism design, and incorporates holistic allocation of decentralized agents' discretion opportunities into a systematic optimization framework.

Table 1: *Central Mechanism Approaches*

	Centralized Approach	**Decentralized Approach**	**Our Hybrid Approach**
System's Performance	Considered	Not Considered	Considered
Agents' Preferences	Not Considered	Considered	Considered
System	Dictatorship	No Intervention	Partially Intervenes
Agents	Committed	No Commitment	No Commitment
Decision Making Process	Fast and Efficient	Takes Time and Effort	Moderate
Common Examples	**Assignment Problem**	**Recommender Systems**	**Bilevel Optimization**

In this research, we advocate for a more flexible recommendation scheme that is neither fully centralized nor decentralized; yet minimize participant effort, enable a quick time to match, and facilitate systematic resource allocation. It must also privilege agent discretion and privacy, and it should increase participation opportunities. We study the role of the central mechanism as a moderator who makes personalized recommendations of limited resources to agents.

In order to do so, we model the problem as a bilevel optimization problem where the system moves first and agents pick their choices based on what systems offers. We explore different solution methods and heuristics to achieve higher performances in terms of solution time. However, the outcomes of the heuristics are slightly different. Then we compare our results with the well-known Gale Shapely algorithm [29].

2 Related literature

2.1 Matching problem

Matching problems are a well-studied area with numerous applications. Although the differed acceptance algorithm [29] is the footstone of this work, different problems have different criteria and axiomatic properties (e.g., pareto optimality, stability, strategy proofness, social welfare). And it is not a one-fit all scenario. So, other solution methods (e.g., serial dictatorship, top trading cycles, etc.) have emerged with unique properties that address the needs of those specific problems.

While stability in the matching problem has attracted lots of attentions, few works also consider other aspects of the matching problem. For example, Anshelevich [cite] argues that a stable match might not necessarily be a socially desirable outcome. It is backed up by the fact that stable matching problem only considers the ranking of agents and thus social welfare is not guaranteed to be optimal. In another work, Kadama and Kojima slightly modified the definition of the stability under existence of the distributional constraints (i.e., regional caps). One of the application of their model is allocation of medical match market of doctors and hospitals when the government enforces certain distributional constraints to overcome geographical imbalance. Although their main goal is still the stability, they allow certain blocking pairs to remain in what so called stable match because otherwise it may violate the caps. This is very similar to our work because the capacity constraints are imposed by a central system (global entity or enterprise). However this central system does not have any control over the choices of agents and thus no recommendation has been made to agents.

Also this central system has no incentive to match certain agents.

2.2 Sharing economy

Existing sharing economy descriptive studies include qualitative studies [8, 9, 17, 19, 37, 41, 47], and econometric models [18, 26].

Prescriptive mathematical models to guide decisions in sharing economy is a new area of research. In [35], pricing and quality decisions are decided for a monopolist manufacturer's physical resource that can be used by its primary owner or made available via a sharing economy. In [10], an equilibrium model is developed for a two-sided market in which agents may not be able to find a alternative (and vice versa). Both consider a decentralized approach. An emerging area of sharing economy research is dynamic ride-sharing, which matches drivers with riders to share one-time trips [2, 28, 38, 42, 58, 63, 64] and peer-to-peer car sharing [33, 34, 56, 57]; all consider either a fully centralized or fully decentralized approach.

Multiple agents with discretion capabilities concerning recommendations from a central mechanism are unstudied. Related is [62], in which systematic and agent preferences are considered by enforcing dynamic ride-sharing matches to be stable. Also, in [49] a central mechanism recommends a single alternative to a single dispatcher who can accept or reject it. The acceptance probability is represented by an exogenous random variable. Neither incorporate discretion into the optimization model, capture the entire spectrum of discretion levels, nor consider dependencies in systematic performance due to multiple agents' decisions.

2.3 Recommender systems

Vast literature on how to create personalized recommendations exists. However, recommender systems research assumes what is being recommended has an infinite capacity, and does not consider users' selections interdependencies [1, 4, 16, 22, 44, 48, 55]. For example, personalized recommendations of articles to read, movies to stream, or ads to click all can be simultaneously consumed without impacting systematic performance. By contrast, this research seeks to create personalized recommendations to multiple decentralized users whose simultaneous selections impact systematic performance.

2.4 Bilevel optimization

The central mechanism must consider agents' likely selection behavior when making the personalized recommendations, because agents have discretion not to comply with the recommendations they do not prefer. Also, agents' selection is highly influenced by the recommendation. Thus, bi-level programming will be used. Bi-level programming problems, also known as Stackelberg leader-follower games [12, 61], model a hierarchy in which a leader makes decisions that effect the followers' feasible decision set [12, 14, 15]. Bi-level optimization problems are recognized as difficult [21]; even linear bi-level programming is NP-hard [5, 6, 27]. Generalized bi-level optimization approaches in which the followers' decision variables are discrete remains computationally challenging [46, 65, 66].

Bi-level research has mainly focused on developing specialized algorithms for specific problems. Prominent bi-level optimization problems include transportation network design, network interdiction, assortment optimization, and revenue management. The network interdiction problem models the attacker-defender problem and specialized approaches (e.g., decomposition approaches and step inequalities) have been shown effective in solving many different variants of this problem [20, 43, 54, 59]. In transportation network design problems, the network design and its capacities are leader decisions; followers solve shortest path problems [13, 23, 32, 45]. Assortment optimization determines what product assortment to offer to an aggregated set of shoppers, who then make consumer selection choices from the recommended assortment [39]. In all problem types, a single aggregate leader decision is made and shared among the followers. For example, all users make routing or interdiction decisions using the same shared network, and all consumers make purchasing selections from the same product assortment. Capacity constraints of a single shared assortment have been considered in assortment optimization [24, 31, 52]. Revenue management research considers what assortment of products (e.g., flight tickets) to display to what segment of users at varying times to sell remaining capacity of a finite resource [30, 40, 67, 68]. Existing revenue management work assumes demand selections occur sequentially and recommendations are independent of each other. Similarly, dynamic assortment optimization considers personalized recommendations of finite capacities to sequentially arriving customers [11, 36, 53, 60]. In contrast, this work makes multiple, *simultaneous* personalized recommendations of finite capacity alternatives.

Existing work in bi-level optimization either models the leader problem as deciding (i) a single aggregate recommendation, or (ii) multiple independent recommendations. The bi-level models proposed in this research are innovative because the leader problem has to make multiple personalized recommendations of finite capacity alternatives. Specialized solution approaches are required because the personalized recommendations cannot be made independently, as systematic performance is dependent on multiple agents' selection outcomes.

3 Model

The bilevel optimization framework consists of a central mechanism as a leader and agents as followers. The central mechanism leads by deciding what personalized set of alternatives to recommend to multiple decentralized independent agents. The agents' utility-maximizing selection processes follow from the personalized recommendations. The model considers two disjoint sets: $j \in S$, the set of agents (which have discretion capabilities), and $i \in R$, the set of alternatives, where $R = D \cup N$. The set D consists of alternatives; the set N consists of the no-choice selections, and $D \cap N = \emptyset$. The recommendations are made simultaneously to all agents and a single period model is presented. Alternatives are assumed to have a capacity of 1. Other notations are as follows.

Decision Variables:

- x_{ij} 1 if the central mechanism recommends alternative $i \in R$ to agent $j \in S$; 0 o.w.
- y_{ij} 1 if agent $j \in S$ selects alternative $i \in R$; 0 o.w.
- z_i 1 if more than one agent selects alternative $i \in D$ (duplicate); 0 o.w.
- w_i 1 if no agents select alternative $i \in D$ (full rejection); 0 o.w.

Parameters:

- C_{ij} Systematic benefit of recommending alternative $i \in R$ to agent $j \in S$.
- u_{ij} Agent $j \in S$ utility of alternative $i \in R$.
- a_i Maximum number of agents recommended alternative $i \in R$, where $a_i = |S|$ if $i \in N$, and $1 \leq a_i \leq |S|$ if $i \in R$.
- b_j Number of alternatives (including no-choice) recommended to agent $j \in S$.
- m_j Number of alternatives (including no-choice) agent $j \in S$ can select.
- d_i Duplication penalty if alternative $i \in D$ receives more than one selection.
- r_i Full rejection penalty if alternative $i \in D$ is not selected by all agents.

Bi-Level Optimization Formulation:

$$\max \sum_{i\in R}\sum_{j\in S} C_{ij}y_{ij} - \sum_{i\in D} d_i z_i - \sum_{i\in D} r_i w_i \tag{1}$$

$$s.t. \sum_{j\in S} x_{ij} \le a_i \quad \forall i \in R \tag{2}$$

$$\sum_{i\in R} x_{ij} \le b_j \quad \forall j \in S \tag{3}$$

$$\sum_{j\in S} y_{ij} \le |D|z_i + 1 \quad \forall i \in D \tag{4}$$

$$1 - \sum_{j\in S} y_{ij} \le w_i \quad \forall i \in D \tag{5}$$

$$x_{ij} \in \{0,1\} \quad \forall i \in R, j \in S \tag{6}$$

$$z_i \in \{0,1\}; w_i \in \{0,1\} \quad \forall i \in D \tag{7}$$

$$\max \sum_{i\in R}\sum_{j\in S} u_{ij}y_{ij} \tag{8}$$

$$s.t. \quad y_{ij} \le x_{ij} \qquad \forall i \in R, j \in S \tag{9}$$

$$\sum_{i\in R} y_{ij} = m_j \quad \forall j \in S \tag{10}$$

$$y_{ij} \in \{0,1\} \quad \forall i \in R, j \in S \tag{11}$$

For a better grasp on the bilevel formulation of this problem, one can consider the network representation of alternatives and agents in figure 1, where the system maximizes the flow from node s to node t in the upper level problem. Maximizing flows backwards from node t to node s are the lower level problem.

The central mechanism's leader problem is captured in (1)-(7). The objective function (1) is to maximize the expected systematic benefit by recommending sets of alternatives to multiple agents. It captures the dependencies among agents' selections by accounting for penalties of duplicate and fully rejected alternatives, which are a function of the agents' decisions (followers). Constraints (2) enforce at most a_i agents can be recommended alternative $i \in D$. Constraints (3) enforce b_j alternatives (including no-choice) are recommended to agent j. Constraints (4) and (5) capture the dependency among multiple agents, whether a alternative is selected by more than one agent (duplicate); or whether a alternative is not selected by any agents (full rejection), respectively.

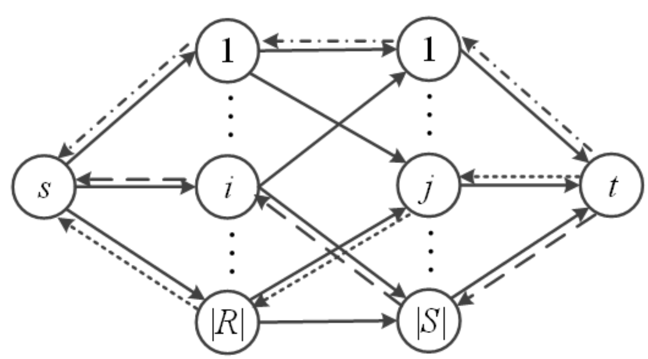

Figure 1: *Example of the network structure. Solid arcs: leader's established arcs (centralized system recommendations), Dotted arcs: followers' final flow (supply users choices)*

The agents' follower problems are captured in (8)-(11). The agents are rational decision makers who make selections based on maximizing their own perceived utility of alternatives in the personalized recommendations (8), selecting the top m_j alternatives in descending order ranked on u_{ij} values. Constraints (9) enforce their selections to be a subset of system's recommendation. In this model, no-choice set $N \in R$ represents discretion in which agents do not have to select all alternatives. The leader and the followers decisions variables are binary in constraints (6)-(7), and

(11) , respectively.

3.1 Solution method

The proposed bi-level framework is a discrete linear bilevel programming problem (DL-BLPP) and can be solved with off the shelf solver [7]. However, the solution time is increasing exponentially and thus become intractable for large scale problems. See figure2.

Due to integer values for m_j, the lower level problem is unimodular and hence the integrality for lower level variables can be relaxed to continuous. (i.e., constraints (11) are relaxed to $0 \leq y_{ij} \leq 1$). This makes the problem to be a discrete-continuous linear bilevel programing problem (DCL-BLPP) which is slightly easier to solve Although algorithms exists to deal with these types of problems [7]. But challenge in solving large scale problems remains the same.

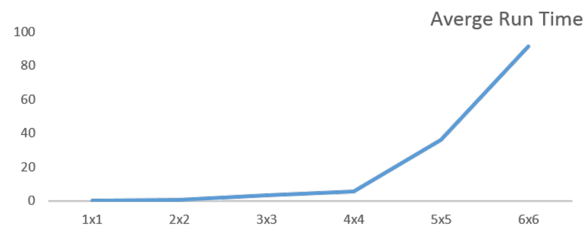

Figure 2: *Average runtime for the DL-BLPP as a function of the problem size*

The outcome of this optimization is two matrices: a recommendation matrix X and a agent selection matrix Y. Note that multiple optimal solutions exits.

3.2 Heuristic 1, projection method

One way of capturing this alignment in the model is employ the projection method that projects the agents' j's utility vector $\vec{U_j}$ onto the systematic benefit vector $\vec{C_j}$ associated with each agent j. Therefore, a projection matrix P captures the alignment of U on C by the well known projection function in (12).

$$\vec{P_j} = Proj_{\vec{C_j}} \vec{U_j} = \frac{\vec{U_j}.\vec{C_j}}{||\vec{C_j}||^2} \vec{C_j} \tag{12}$$

Figure 3 illustrates an example of this alignment concept where $\vec{C}_1$ and $\vec{U}_1$ are completely aligned to each other whereas $\vec{C}_2$ and $\vec{U}_U$ are not. The projection matrix P represents a compact information of U and C together where the magnitude by which the vector $\vec{U_j}$ is aligned to vector $\vec{C_j}$

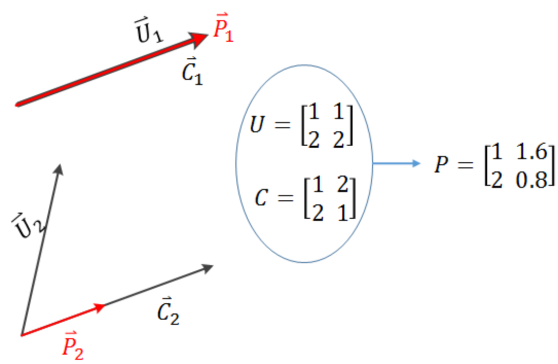

Figure 3: *Example of two vectors*

Single Level Optimization Formulation:

Applying this technique, we are able to combine both levels into one single level optimization as shown below.

$$\max \sum_{i \in R} \sum_{j \in S} P_{ij} y_{ij} - \sum_{i \in D} d_i z_i - \sum_{i \in D} r_i w_i \tag{13}$$

$$s.t. \sum_{j \in S} x_{ij} \leq a_i \quad \forall i \in R \tag{14}$$

$$\sum_{i \in R} x_{ij} \leq b_j \quad \forall j \in S \tag{15}$$

$$\sum_{j \in S} y_{ij} \leq |D| z_i + 1 \quad \forall i \in D \tag{16}$$

$$1 - \sum_{j \in S} y_{ij} \leq w_i \quad \forall i \in D \tag{17}$$

$$y_{ij} \leq x_{ij} \qquad \forall i \in R, j \in S \tag{18}$$

$$\sum_{i \in R} y_{ij} = m_j \quad \forall j \in S \tag{19}$$

$$x_{ij} \in \{0,1\}; y_{ij} \in \{0,1\} \quad \forall i \in R, j \in S \tag{20}$$

$$z_i \in \{0,1\}; w_i \in \{0,1\} \quad \forall i \in D \tag{21}$$

The single level formulation is similar to the bilevel formulation except that in the single level formulation, the follower objective function (8) is not present. Also, the upper level objective function in the bilevel formulation (1) is replaced with the objective function (13) which maximizes the benefit by applying the projection method (12) and yet captures the dependencies among agents' selections by accounting for penalties of duplicate and fully rejected alternatives. All other constraints remain the same as before and are presented in a single level.

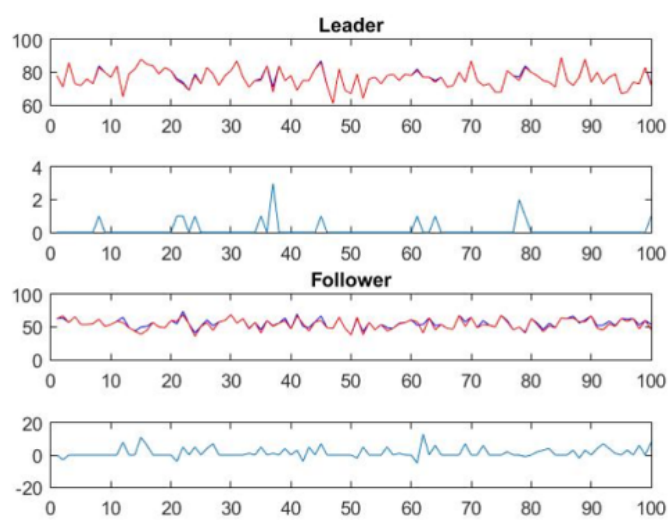

Figure 4: *Bilevel performance vs single level performance. Charts from top to bottom: (1) leader's bilevel objective function in dark blue, leader's single level objective funtion in red. (2) difference between single and bilevel optimizagtion. (3) followers's bilevel objective function in dark blue, leader's single level objective funtion in red. (4) the difference between the two.*

The proposed single level framework is a mixed integer programming (MIP) and numerous algorithms and solvers exist that solves the MIP very efficiently in reasonable time. We used branch and bound technique introduced in [25] to solve this problem. Experiments of different sizes have been tested and the results are compared to the results of the bilevel optimization. A comparison of hundred iterations of recommendations of 5 alternatives to 5 agents is shown in the figure (4) where the U and the C matrices are randomly generated. However, as illustrated in the figure, the leader's objective function of the single level optimization is near optimal. Moreover, inconsistencies occur for the follower's objective function in single level solution (i.e., some times lower and some times higher). Therefore, we study other heuristics and compare the performances.

4 Heuristic 2, Greedy Method

We also tested an alternative to the bilevel optimization framework, which attempted to reconcile deferred acceptance algorithms like Gale Shapley, described in [29], with the greedy algorithm paradigm in order to maximize system performance at the sacrifice of social welfare utility. It should be noted that in doing so the stable matching requirements held as being a central property in deferred acceptance algorithms are not a concern for ours. Also note that the susceptibility of our greedy algorithm to manipulation was not evaluated, in order to focus efforts on examining the algorithms potential viability.

In order to evaluate the greedy algorithms effectiveness, we decided to use the Gale Shapley algorithm as a baseline and compare its system performance and social welfare to that of our algorithm. However, we had to adapt Gale Shapley to accommodate for many to many matching.

4.1 Greedy Algorithm for Many to Many Matching

The user matrix represents the preferences of the agents being proposed to. The system matrix represents the preferences of the agents doing the proposing. Proposal order is determined by taking the sum of the utilities for each column in the user matrix, where this is basically determining individual system desirability for all users, and from this a list of systems is created, sorted by desirability in decending order. We then iterate over this list, and each system then picks their highest rated (available) users, up to the maximum allowed matches.

Note that each system and user can have at most the maximum allowed matches at any given time, but each system and user does not necessarily have a complete matching at the end of the algorithm. This case is more thoroughly examined in the Gale Shapley algorithm discussion.

4.2 Example

Say we have a 4 by 4 syatem and user matrix, and wish to apply our new algorithm. Let us first declare that the maximum matches we are seaching for is 2. Then let us declare the system and user preference matrices, where systems are E, F, G, and H and are on the horizonal axis, and users are A, B, C, and D and are on the vertical axis. The notation in the matrices describes an agents preference to another agent, where the first letter is the agent doing the considering, the second letter is the agent being considered, and the subscript is the utility the first agent gives the second.

$$SystemMatrix \begin{bmatrix} EA_3 & FA_2 & GA_2 & HA_2 \\ EB_4 & FB_3 & GB_4 & HB_3 \\ EC_2 & FC_4 & GC_1 & HC_1 \\ ED_1 & FD_1 & GD_3 & HD_4 \end{bmatrix}, UserMatrix \begin{bmatrix} AE_3 & AF_1 & AG_4 & AH_2 \\ BE_4 & BF_2 & BG_1 & BH_3 \\ CE_1 & CF_4 & CG_2 & CH_3 \\ DE_2 & DF_1 & DG_4 & DH_3 \end{bmatrix}$$

The first step is to compute system desirability as defined by the users, and so we sum each column in the user matrix and produce the set:

$$UnsortedProposalSet \begin{bmatrix} E_{10} & F_8 & G_{11} & H_{11} \end{bmatrix}$$

Which we then sort in decending order to give us the order the systems will propose to users.

$$SortedProposalSet \begin{bmatrix} G_{11} & H_{11} & E_{10} & F_8 \end{bmatrix}$$

Then we go over every system based on this order, and each one then proposes to its top users based on the utilities in the system matrix until it has all the matches required (in this case, 2). A proposal is accepted by a user only if that user has less than the maximum matches already. Otherwise the system must move on to its next most preferred user.

After all systems have finished proposing, we are left with this result matrix:

$$ResultMatrix \begin{bmatrix} AE_1 & AF_1 & AG_0 & AH_0 \\ BE_0 & BF_0 & BG_1 & BH_1 \\ CE_1 & CF_1 & CG_0 & CH_0 \\ DE_0 & DF_0 & DG_1 & DH_1 \end{bmatrix}$$

Where the letters now describe the match and the subscript describes if the match is true or not (1 if true, 0 if false).

We now can take system performance and social welfare utility by taking the double sum of the dot product bteween the result matrix and system and user matrices, respectively. In this case, system performance is computed to be 25, and social welfare utility to be 20.

4.3 Gale Shapley Algorithm Adaptation for Many to Many Matching

At it's core we use a standard implementation of the Gale Shapley algorithm, which ensures a stable one to one matching between two agent sets. The difference here is that we must also account for many to many matching, which is done by allowing each proposer to make at most the maximum allowed match proposals at each iteration.

An important case to mention is when a proposer tried to match to a proposee that is alread at the maximum allowed matches and tie breaking needs to happen. This is done by taking the old proposer with the lowest utility beaten by the new proposer out of the proposee match list and adding them back to the match queue. This, combined with the rest of the Gale Shapley algorithm, means the possibility exists that a full many to many matching for any two system and user matrices does not necessarily exist.

4.4 Example

For consistencies sake, let us take the same system and user matrices used in the greedy algorithm example, and define the system agents as the proposers. Note the notation within the matrices is the same.

$$SystemMatrix \begin{bmatrix} EA_3 & FA_2 & GA_2 & HA_2 \\ EB_4 & FB_3 & GB_4 & HB_3 \\ EC_2 & FC_4 & GC_1 & HC_1 \\ ED_1 & FD_1 & GD_3 & HD_4 \end{bmatrix}, UserMatrix \begin{bmatrix} AE_3 & AF_1 & AG_4 & AH_2 \\ BE_4 & BF_2 & BG_1 & BH_3 \\ CE_1 & CF_4 & CG_2 & CH_3 \\ DE_2 & DF_1 & DG_4 & DH_3 \end{bmatrix}$$

Let us define the match queue at the first iteration to contain all systems, E, F, G and H. E matches to A and B, F matches to B and C, and then G matches to A and D but not B, since B prefers E and F over G. Then H matches to D and proposes to B, and since B prefers H over F, it replaces F with H and F is added back to the proposal queue. Now we are on the second iteration, where only F needs to find another match. F proposes to A, but A prefers it's two matches, E and G, over F. F then proposes to D, but like before D prefers it's two current matches, G and H, over F. This then leaves F with no more users to propose to and the algorithm ends, and also gives us an example of when a complete many to many matching is not found using Gale Shapley. The result matrix is then:

$$ResultMatrix \begin{bmatrix} AE_1 & AF_0 & AG_1 & AH_0 \\ BE_1 & BF_0 & BG_0 & BH_1 \\ CE_0 & CF_1 & CG_0 & CH_0 \\ DE_0 & DF_0 & DG_1 & DH_1 \end{bmatrix}$$

Where the letters now describe the match and the subscript describes if the match is true or not (1 if true, 0 if false).

System performance and social welfare utility, when computed as defined in the greedy algorithm example, is now 23 and 25, respectively. Compared to the utilities of that example (25 and 20, respectively), we can see how the greedy algorithm boosts system performance at the cost of social welfare utility in this case.

4.5 Experiment Results

In order to examine the performance and result trends of the greedy algorithm when compared to Gale Shapley, we conducted several large scale tests with random matrices of increasing size and match number requirements.

				Our Model	Our Model	Gale Shapley	Gale Shapley
Test	Alternatives	Agents	Max. Matches	Sys. Benefit	Soc. Welfare	Sys. Benefit	Soc. Welfare
1	4	4	2	24	19	23	20
2	4	4	2	24	18	23	20
3	5	5	2	40	29	38	31
4	6	6	3	80	61	78	66
5	7	7	3	112	80	109	91
6	8	8	4	186	140	180	154
7	9	9	4	243	175	234	195
8	10	10	5	361	266	353	299

5 Conclusion and future steps

This research studies the case where system performance is a major concern when matches are assigned. However, this does not come at the price of ignoring agent's preferences in recommendations. So, the recommendations are made to agents considering their preferences under the priority of system's performance. Comparing the results of our greedy algorithm to DA stable matching algorithm, we conclude that the system performance in our greedy approach is more than or equal to DA stable matching and the social welfare in our greedy approach is less than or equal to the DA stable marching algorithm. For future works, we want to calculate the bound on this comparison and measure the maximum possible difference in system performance as well as social welfare in both algorithms. In order to do so, we need to consider worse case scenarios. Moreover, in this study, we set the alternatives to propose first to guarantee optimality for the system. However, to fully harness the characteristics of this problem, agent-proposing method still needs to be studied as well.

References

[1] Adomavicius, G., and Tuzhilin, A., "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions," *IEEE Transactions on Knowledge and Data Engineering*, 17, 6, 734–749 (2005).

[2] Agatz, N., Erera, A., Savelsbergh, M., and Wang, X., "Optimization for dynamic ride-sharing: a review," *European Journal of Operational Research*, 223, 2, 295–303 (2012).

[3] Allen, D., "The sharing economy," *Review-Institute of Public Affairs*, 67, 3, 24 (2015).

[4] Amatriain, X., Jaimes, A., Oliver, N., and Pujol, J. M., "Data mining methods for recommender systems," In *Recommender Systems Handbook*, 39–71. Springer (2011).

[5] Audet, C., Hansen, P., Jaumard, B., and Savard, G., "Links between linear bilevel and mixed 0–1 programming problems," *Journal of Optimization Theory and Applications*, 93, 2, 273–300 (1997).

[6] Bard, J. F., *Practical bilevel optimization: algorithms and applications*, volume 30, Springer Science & Business Media (1998).

[7] Bard, J. F., *Practical bilevel optimization: algorithms and applications*, volume 30, Springer Science & Business Media (2013).

[8] Bardhi, F., and Eckhardt, G. M., "Access-based consumption: the case of car sharing," *Journal of Consumer Research*, 39, 4, 881–898 (2012).

[9] Belk, R., "You are what you can access: sharing and collaborative consumption online," *Journal of Business Research*, 67, 8, 1595–1600 (2014).

[10] Benjaafar, S., Kong, G. C., Li, X., and Courcoubetis, C., "Peer-to-peer product sharing: Implications for ownership, usage and social welfare in the sharing economy," *Working Paper* (2015).

[11] Bernstein, F., Kök, A. G., and Xie, L., "Dynamic assortment customization with limited inventories," *Manufacturing & Service Operations Management*, 17, 4, 538–553 (2015).

[12] Bracken, J., and McGill, J. T., "Mathematical programs with optimization problems in the constraints," *Operations Research*, 21, 1, 37–44 (1973).

[13] Brotcorne, L., Labbé, M., Marcotte, P., and Savard, G., "A bilevel model for toll optimization on a multicommodity transportation network," *Transportation Science*, 35, 4, 345–358 (2001).

[14] Candler, W., and Norton, R., *Multi-level programming and development policy*, The World Bank (1977).

[15] Colson, B., Marcotte, P., and Savard, G., "An overview of bilevel optimization," *Annals of operations research*, 153, 1, 235–256 (2007).

[16] Condliff, M. K., Lewis, D. D., Madigan, D., and Posse, C., "Bayesian mixed-effects models for recommender systems," In *ACM SIGIR99 Workshop on Recommender Systems: Algorithms and Evaluation*, volume 15. Citeseer (1999).

[17] Corciolani, M., and Dalli, D., "Gift-giving, sharing and commodity exchange at Bookcrossing.com. New insights from a qualitative analysis," *Management Decision*, 52, 4, 7–7 (2014).

[18] Cullen, Z., and Farronato, C., "Outsourcing tasks online: Matching supply and demand on peer-to-peer internet platforms," *Working Paper* (2014).

[19] Deakin, E., Frick, K. T., and Shively, K. M., "Markets for dynamic ridesharing?," *Transportation Research Record: Journal of the Transportation Research Board*, 2187, 1, 131–137 (2010).

[20] Delgadillo, A., Arroyo, J. M., and Alguacil, N., "Analysis of electric grid interdiction with line switching," *IEEE Transactions on Power Systems*, 25, 2, 633–641 (2010).

[21] Dempe, S., and Franke, S., "On the solution of convex bilevel optimization problems," *Computational Optimization and Applications*, 63, 3, 685–703 (2016).

[22] Ekstrand, M. D., Riedl, J. T., and Konstan, J. A., "Collaborative filtering recommender systems," *Foundations and Trends in Human-Computer Interaction*, 4, 2, 81–173 (2011).

[23] Farahani, R. Z., Miandoabchi, E., Szeto, W. Y., and Rashidi, H., "A review of urban transportation network design problems," *European Journal of Operational Research*, 229, 2, 281–302 (2013).

[24] Feldman, J. B., and Topaloglu, H., "Capacity constraints across nests in assortment optimization under the nested logit model," *Operations Research*, 63, 4, 812–822 (2015).

[25] Fourer, R., Gay, D., and Kernighan, B., *Ampl*, volume 117, Boyd & Fraser Danvers, MA (1993).

[26] Fradkin, A., Grewal, E., Holtz, D., and Pearson, M., "Bias and reciprocity in online reviews: Evidence from field experiments on airbnb," In *Proceedings of the Sixteenth ACM Conference on Economics and Computation*, 641–641. ACM (2015).

[27] Frangioni, A., "On a new class of bilevel programming problems and its use for reformulating mixed integer problems," *European Journal of Operational Research*, 82, 3, 615–646 (1995).

[28] Furuhata, M., Dessouky, M., Ordóñez, F., Brunet, M.-E., Wang, X., and Koenig, S., "Ridesharing: the state-of-the-art and future directions," *Transportation Research Part B: Methodological*, 57, 28–46 (2013).

[29] Gale, D., and Shapley, L. S., "College admissions and the stability of marriage," *The American Mathematical Monthly*, 69, 1, 9–15 (1962).

[30] Gallego, G., Ratliff, R., and Shebalov, S., "A general attraction model and sales-based linear program for network revenue management under customer choice," *Operations Research*, 63, 1, 212–232 (2014).

[31] Gallego, G., and Topaloglu, H., "Constrained assortment optimization for the nested logit model," *Management Science*, 60, 10, 2583–2601 (2014).

[32] Gao, Z., Wu, J., and Sun, H., "Solution algorithm for the bi-level discrete network design problem," *Transportation Research Part B: Methodological*, 39, 6, 479–495 (2005).

[33] Hampshire, R. C., and Gaites, C., "Peer-to-peer carsharing," *Transportation Research Record: Journal of the Transportation Research Board*, 2217, 1, 119–126 (2011).

[34] Hampshire, R. C., and Sinha, S., "A simulation study of peer-to-peer carsharing," In *2011 IEEE Forum on Integrated and Sustainable Transportation System (FISTS)*, 159–163 (2011).

[35] Jiang, B., and Tian, L., "Collaborative Consumption: Strategic and Economic Implications of Product Sharing," *Working Paper, available at SSRN 2561907* (2016).

[36] Johari, R., Kamble, V., and Kanoria, Y., "Know your customer: Multi-armed bandits with capacity constraints," *Working Paper available at arXiv preprint arXiv:1603.04549* (2016).

[37] Johnson, M., Herrmann, A., and Huber, F., "Growth through product-sharing services," *Journal of Service Research*, 1, 167–177 (1998).

[38] Kleiner, A., Nebel, B., Ziparo, V. A., and Srl, A., "A mechanism for dynamic ride sharing based on parallel auctions," In *Procedings of the Twenty-Second International Joint Conference on Artificial Intelligence*, 266–272 (2011).

[39] Kök, A. G., Fisher, M. L., and Vaidyanathan, R., "Assortment planning: Review of literature and industry practice," In *Retail supply chain management*, 175–236. Springer (2015).

[40] Kunnumkal, S., and Topaloglu, H., "A new dynamic programming decomposition method for the network revenue management problem with customer choice behavior," *Production and Operations Management*, 19, 5, 575–590 (2010).

[41] Lewis, A., and Simmons, M., "P2P carsharing service design: informing user experience development," Master's thesis, Sustainable Product-Service System Innovation, School of Engineering, Blekinge Institute of Technology, Karlskrona, Sweden (2012).

[42] Li, B., Krushinsky, D., Reijers, H. A., and Van Woensel, T., "The share-a-ride problem: People and parcels sharing taxis," *European Journal of Operational Research*, 238, 1, 31–40 (2014).

[43] Lim, C., and Smith, J. C., "Algorithms for discrete and continuous multicommodity flow network interdiction problems," *IIE Transactions*, 39, 1, 15–26 (2007).

[44] Lops, P., De Gemmis, M., and Semeraro, G., "Content-based recommender systems: State of the art and trends," In *Recommender Systems Handbook*, 73–105. Springer (2011).

[45] Marcotte, P., "Network design problem with congestion effects: A case of bilevel programming," *Mathematical programming*, 34, 2, 142–162 (1986).

[46] Moore, J. T., and Bard, J. F., "The mixed integer linear bilevel programming problem," *Operations Research*, 38, 5, 911–921 (1990).

[47] Murphy, C., "Shared Mobility and the Transformation of Public Transit," *The National Academies Transportation Research Board TCRP J-11/TASK 21* (2016).

[48] Pazzani, M. J., and Billsus, D., "Content-based recommendation systems," In *The adaptive web*, 325–341. Springer (2007).

[49] Powell, W. B., Towns, M. T., and Marar, A., "On the value of optimal myopic solutions for dynamic routing and scheduling problems in the presence of user noncompliance," *Transportation Science*, 34, 1, 67–85 (2000).

[50] Resnick, P., and Varian, H. R., "Recommender systems," *Communications of the ACM*, 40, 3, 56–58 (1997).

[51] Roughgarden, T., *Selfish routing and the price of anarchy*, MIT Press Cambridge (2005).

[52] Rusmevichientong, P., Shen, Z. M., and Shmoys, D. B., "Dynamic assortment optimization with a multinomial logit choice model and capacity constraint," *Operations Research*, 58, 6, 1666–1680 (2010).

[53] Sauré, D., and Zeevi, A., "Optimal dynamic assortment planning with demand learning," *Manufacturing & Service Operations Management*, 15, 3, 387–404 (2013).

[54] Scaparra, M. P., and Church, R. L., "A bilevel mixed-integer program for critical infrastructure protection planning," *Computers & Operations Research*, 35, 6, 1905–1923 (2008).

[55] Schafer, J. B., Frankowski, D., Herlocker, J., and Sen, S., "Collaborative filtering recommender systems," In *The adaptive web*, 291–324. Springer (2007).

[56] Shaheen, S. A., and Cohen, A. P., "Carsharing and personal vehicle services: worldwide market developments and emerging trends," *International Journal of Sustainable Transportation*, 7, 1, 5–34 (2013).

[57] Steininger, K. W., and Bachner, G., "Extending car-sharing to serve commuters: an implementation in Austria," *Ecological Economics*, 101, 64–66 (2014).

[58] Stiglic, M., Agatz, N., Savelsbergh, M., and Gradisar, M., "Making dynamic ride-sharing work: The impact of driver and rider flexibility," *Transportation Research Part E: Logistics and Transportation Review*, 91, 190–207 (2016).

[59] Sullivan, K. M., Smith, J. C., and Morton, D. P., "Convex hull representation of the deterministic bipartite network interdiction problem," *Mathematical Programming*, 145, 1-2, 349–376 (2014).

[60] Topaloglu, H., "Joint stocking and product offer decisions under the multinomial logit model," *Production and Operations Management*, 22, 5, 1182–1199 (2013).

[61] Von Stackelberg, H., *The theory of the market economy*, Oxford University Press (1952).

[62] Wang, X., Agatz, N., and Erera, A., "Stable matching for dynamic ride-sharing systems," *ERIM Report Series Reference No. ERS-2015-006-LIS* (2014).

[63] Winter, S., and Nittel, S., "Ad hoc shared ride trip planning by mobile geosensor networks," *International Journal of Geographical Information Science*, 20, 8, 899–916 (2006).

[64] Xing, X., Warden, T., Nicolai, T., and Herzog, O., "SMIZE: a spontaneous ride-sharing system for individual urban transit," In *Multiagent System Technologies*, 165–176. Springer (2009).

[65] Xu, P., *Three essays on bilevel optimization algorithms and applications*, PhD thesis, Iowa State University (2012).

[66] Zeng, B., and An, Y., "Solving bilevel mixed integer program by reformulations and decomposition," *Optimization On-line* (2014).

[67] Zhang, D., and Adelman, D., "An approximate dynamic programming approach to network revenue management with customer choice," *Transportation Science*, 43, 3, 381–394 (2009).

[68] Zhang, D., and Cooper, W. L., "Revenue management for parallel flights with customer-choice behavior," *Operations Research*, 53, 3, 415–431 (2005).