Selling Tomorrow's Bargains Today

Melika Abolhassani
University of Maryland
College Park, MD, USA
melika@cs.umd.edu

Hossein Esfandiari
University of Maryland
College Park, MD, USA
hossein@cs.umd.edu

MohammadTaghi Hajiaghayi
University of Maryland
College Park, MD, USA
hajiagha@cs.umd.edu

Hamid Mahini
University of Maryland
College Park, MD, USA
hmahini@cs.umd.edu

David Malec
University of Maryland
College Park, MD, USA
dmalec@umiacs.umd.edu

Aravind Srinivasan
University of Maryland
College Park, MD, USA
srin@cs.umd.edu

ABSTRACT

Consider a good (such as a hotel room) which, if not sold on time, is worth nothing to the seller. For a customer who is considering a choice of such goods, their prices may change dramatically by the time the customer needs to use the good; thus a customer who is aware of this fact might choose to gamble, delaying buying until the last moment in the hopes of better prices. While this gamble can yield large savings, it also carries much risk. However, a coordinator can offer customers a compromise between these extremes and benefits in aggregate. Here we explore how a coordinator might profit from forecasts of such future price fluctuations. Our results can be used in a general setting where customers buy products or services in advance and where market prices may significantly change in the future.

We model this as a two-stage optimization problem, where the coordinator first agrees to serve some buyers, and then later executes all agreements once the final values have been revealed. Agreements with buyers consist of a set of acceptable options and a price where the details of agreements are proposed by the buyer. We investigate both the profit maximization and loss minimization problems in this setting. For the profit maximization problem, we show that the profit objective function is a non-negative submodular function, and thus we can approximate its optimal solution within an approximation factor of 0.5 in polynomial time. For the loss minimization problem, we first leverage a sampling technique to formulate our problem as an integer program. We show that there is no polynomial algorithm to solve this problem optimally, unless $P = NP$. In addition, we show that the corresponding integer program has a high integrality gap and it cannot lead us to an approximation algorithm via a linear-programming relaxation. Nevertheless, we propose a bicriteria-style approximation that gives a constant-factor approximation to the minimal loss by allowing a fraction of our options to overlap. Importantly, however, we show that our algorithm provides a strong, uniform bound on the amount the overlap per options. We propose our algorithm by rounding the optimal solution of the relaxed linear program via a novel dependent-rounding method.

Appears in: *Proceedings of the 14th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2015), Bordini, Elkind, Weiss, Yolum (eds.), May 4–8, 2015, Istanbul, Turkey.*

Categories and Subject Descriptors

F.2.2 [**Theory of Computation**]: Analysis of Algorithms and Problem Complexity—*Nonnumerical Algorithms and Problems*

General Terms

Algorithms

Keywords

two-stage optimization; matching; efficient algorithm

1. INTRODUCTION

Let us start with an example of a basic source of uncertainty in E-commerce. Google offers high-quality free services for retaining Internet users and makes over 96% of its revenue from advertisers by selling users' attention to them. For this purpose, Google provides its AdWords system, an online auction-based advertising system, that lets advertisers bid on keywords for showing their ads in Google's search results. Advertisers can participate in Google's on-line AdWords auction and bid on their keywords. However, the cost-per-click (CPC) amount that an advertiser should pay when users click on its ads depends heavily on the online demand and competitors' bids, and thus a (near-)optimal bidding strategy is not clear to advertisers at bidding time. This unknown behavior of prices may force advertisers to take too much risk at bidding time. Many risk-averse advertisers prefer to avoid such risk, and attempt to sign a contract which guarantees an appropriate number of clicks for a fixed price.

This phenomenon arises more generally due to uncertainty such as uncertain future demand, uncertainty in future costs, and uncertain competitors' behavior. While we started with an application in the online advertising industry, we continue with another example of this phenomenon in the hotel-reservation industry.[1] Consider a family that decides, on Monday, that they would like to go on vacation the following weekend. Perhaps they do some research, and find a convenient location that seems both pleasant and affordable. All that is left for them to do is actually reserve their accommodations. But this involves an interesting dilemma: should they book a room now, or wait until late in the week? Booking now assures them a place to stay that is affordable. On the other hand, many hotels offer last-minute deals, which could save the potential vacationers money if they decide to wait. Unfortunately, the latter

[1] A company in the hotel reservation market has based their business strategy around this phenomenon [13].

carries not only the chance for large savings, but the risk that prices will go up, perhaps even to the point where the vacation becomes impossible.

In this work, we study how a company might profit by offering customers a compromise between these options. While dealing with online prices typically carries too much risk and requires significant effort to appeal to individual customers, a coordinator has the advantage of spreading these risks across many contracts. By expending the effort to collect pricing data and form estimates of future prices, a company could reasonably hope to monetize this advantage by offering customers a reliable contract with an affordable price, while executing the contract when prices are as favorable as possible – while not every contract may be profitable, good price estimates should provide a profit in aggregate.

In fact, this opportunity arises more generally – the key relevant aspects of our examples are uncertain future prices. Thus, one could hope to exploit this sort of future arbitrage when selling stock options, airline tickets, rental cars, event tickets, or any product/service that typically faces price fluctuations. Our goal in this work is to answer this question: given estimates of future prices, what is the best way for an enterprising coordinator to offer contract to buyers?

Two-stage optimization.

We have a coordinator who can provide options from a set H, and who will have a chance to offer these options to a set of potential buyers B. This process, however, takes places in stages: in the first stage, the coordinator negotiates agreements; in the second stage, the prices will be realized, and the coordinator must serve options in the realized *scenario* to fulfill all of the previously made agreements. Each agreement with a buyer $b \in B$ specifies a *pack* $P \subseteq H$ of options that are acceptable to the buyer, and a *value* v_b the buyer must pay. The coordinator may satisfy the agreement by getting any option in the pack to the buyer, and it does not matter which one. The two-stage nature of our problem arises because the coordinator must make binding decisions about what agreements to make *before* the prices are revealed.

First stage: agreements.

The first stage of our optimization problem models the formation of agreements. In our model, all of the buyers arrive at once, and each proposes a pack of options and a price. The value v_b associated with each buyer is the price they propose, and the coordinator may accept a subset of offers[2]. Note that agreements are only formed when an offer is made and the coordinator accepts; therefore, we refer to the set S of buyers the coordinator chooses to form agreements with as the *served set*.

Second stage: execution.

In the second stage, the coordinator must match each buyer $b \in S$ to an option in their associated pack. At this point, the prices are revealed, and the coordinator's problem becomes one of maximum-weight matching. We call the collection of revealed prices a *scenario*, and denote it by I; we denote the full set of possible scenarios by $\mathcal{I}$. We denote the price of option h in scenario I by c_h^I. The I seen in the second stage is drawn according to a probability distribution, and the coordinator has the ability to sample from this distribution.

Objectives.

The coordinator's objective is to maximize *profit*. We denote the profit from a served set S as

$$P(S) = \sum_{b \in S} v_b + \mathrm{E}[\sum_{h \notin \mathcal{M}^I(S)} c_h^I],$$

where $\mathcal{M}^I(S)$ is the cheapest set of options that buyers in S can be matched to in scenario I, and the expectation is over which I occurs. The first term is the profit that is extracted from agreements in S, e.g., set of contracts in the Google advertising example. The second term is the profit that is made by selling the remaining options in the future, e.g., selling through online Google AdWords system.

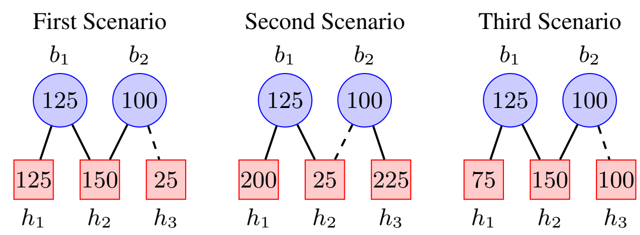

Figure 1: Each graph corresponds to one scenario. The upper vertices show the buyers and the price they are willing to pay. The lower vertices show the options and their realized values in each scenario. The edges indicate the buyers' interest in options. In this example, the best decision is to choose b_2, for which our best matches are shown with dashes.

EXAMPLE 1.1. *In this example, there are two buyers b_1 and b_2, three options h_1, h_2, and h_3, and three possible future scenarios. Each scenario can be represented by a vector of 3 elements indicating the realized values of the three options. Assume the future scenarios are $I_1 = \{125, 250, 25\}$, $I_2 = \{200, 25, 225\}$, and $I_3 = \{75, 150, 100\}$, and they happen with probabilities 0.4, 0.3, and 0.3, respectively. The first buyer is willing to pay a price equal to 125 dollars for being served, while the second buyer will pay 100 dollars. Figure 1 illustrates this example.*

In this case, if we only serve b_1, the best matches to this buyer in future scenarios I_1, I_2, and I_3 are h_1, h_2, and h_1 dollars, respectively. Therefore, our expected profit from choosing only b_1 to serve would be $125 + (0.4(150 + 25) + 0.3(200 + 225) + 0.3(150 + 100)) = 397.5$ dollars. On the other hand, if we only choose b_2, our expected profit would be $100 + (0.4(125 + 150) + 0.3(200 + 225) + 0.3(75 + 150)) = 405$ dollars. Finally, if we choose both buyers to serve, we may no longer be able to serve each buyer with their cheapest feasible option. In the first scenario, the best options for b_1 and b_2 would be h_1 and h_3, respectively, and the remaining value is 150 dollars. Similarly, the remaining values would be 225 and 150 dollars when serve both buyers in the second and third scenarios, respectively. Therefore, the expected total profit from serving both customers would be $125 + 100 + (0.4 \times 150 + 0.3 \times 225 + 0.3 \times 150) = 397.5$ dollars. Thus, our best option is to only serve b_2 for a total profit of 405 dollars, even though her offered price is less than the price b_1 is willing to pay us.

In some applications such as in the hotel-reservation industry, the value c_h^I can be interpreted as the cost of providing option h in scenario I. In these situations, we study the loss minimization problem rather than the profit maximization version. Therefore, we also consider a modified objective that we call *loss*, which has the form

$$L(S) = \sum_{b \in B \setminus S} v_b + \mathrm{E}[\sum_{h \in \mathcal{M}^I(S)} c_h^I].$$

[2]We may use "price" interchangeably with "value" herein.

Note that $L(S)$ is an affine transformation of $P(S)$. Intuitively, the loss objective tries to capture the idea of lost revenue, where we can lose revenue either by choosing not to serve a buyer, or by having to spend to pay for an option.

1.1 Our results

Since buyers may specify any pack of options they like, the marginal value of serving a particular customer becomes hard to quantify – they may conflict with other buyers in complex and arbitrary ways. Nevertheless, we prove that the profit objective function is submodular, and thus a polynomial-time algorithm approximates the optimum value within a factor of 0.5 due to the result of Buchbinder et al. [3].

Theorem (Theorem 2.3) The profit function is submodular.

To prove this theorem, we rewrite $P(S)$ as $\sum_{b\in S} v_b + \mathrm{E}[\sum_{h\in S} c_h^I] - C(S)$ where $C(S)$ is the expected value of the cheapest matching over scenarios or equivalently $C(S) = \sum_{h\in\mathcal{M}^I(S)} c_h^I$. The challenge is to prove that $C(S)$ is supermodular. It is enough to show that it is supermodular for each scenario individually. We first prove a lemma to demonstrate how for sets $S \subseteq T \subseteq B$ we can change a matching of S to a matching of T. Then we apply this lemma to the sets $T \cup \{b\}$ and S to show that the distance between the value of these two matchings $C(T \cup \{b\}) - C(S)$ can be broken into two disjoint matching distances. The first one between $S \cup \{b\}$ and S and the second one between the sets T and S. This gives us an inequality which immediately yields the supermodularity of $C(S)$.

We also study the loss objective function that has been used in the associated literature [16, 1, 11, 2]. The loss objective has the form of "missed" value; note that exactly minimizing loss is equivalent to exactly maximizing profit. Unfortunately, the loss object function is supermodular and does not have the nice structural property of the profit objective function. In Theorem 3.2 we show that it is not possible to optimally minimize the loss function in polynomial time unless $P = NP$. In order to approximate the solution, in Section 4, we show that we can use sampling to construct an integer program that, with high probability, provides a $(1 + O(\epsilon))$-approximation to the loss objective. We call this integer program the second-stage allocation IP (see Figure 2). Unfortunately, this does not directly lead to an approximation via the standard approach of linear-programming relaxations, as we show that the integrality gap of the corresponding linear program is quite high.

Theorem (Theorem 3.3) The integrality gap of the second-stage allocation IP is at least $\Omega(n)$ where the total number of buyers is $O(n)$.

We prove this theorem by constructing an example with $2n$ customers and $O(4^n)$ options in which the fractional solution is 2 and the integral solution cannot be better than $n + 1$. We consider all the values of customers to be equal to 1. The idea is to come up with the right edges and set of scenarios such that no n customers can be satisfied simultaneously. In other words, if we choose a set of n customers, we cannot match them at the same time in at least one scenario. For each set of n customers, we consider a scenario in which all of these customers have only edges to $n - 1$ affordable options, and therefore, we have to miss at least one of them in the contracts. However, we show there exists a solution to the LP which fractionally matches all customers in each instance, and does not have to miss any of them.

The high integrality gap of the aforementioned linear program leads us to consider bicriteria-style approximations; our main result is the following, which provides an approximation to the loss objective by relaxing the matching constraints between buyers and options.

Theorem (Theorem 3.5) Any fractional solution to the second-stage allocation integer program can be rounded to an integral solution while increasing the loss objective value by at most a factor of $1/f$, while ensuring that no option is matched to more than 2 buyers and at most a $\min\{\frac{f}{1-2f}, \frac{1}{2}\}$ fraction of buyers cannot be uniquely matched to an option, for any $0 < f < \frac{1}{2}$.

We propose an integer programming formulation for solving this problem. In order to obtain Theorem 3.5, we first relax the integer program to a linear program (LP), and leverage the sampling technique to propose a polynomial-time algorithm for solving the corresponding linear program. Then, we show how to round the LP-solution using an appropriate dependent rounding. One of the tools we use in rounding the LP-solution is a certain type of bipartite dependent-rounding procedure developed in [9]. In particular, (a) this helps show that with probability one, no option has more than two buyers assigned, and (b) gives us a handle both on the (expected) number of overbooked customers and on the probability of assigning a customer to an option. The "dependence" in the rounding helps with issue (a), while inheriting the property (b) from *independent* rounding schemes.

1.2 Related work

Our problem falls into the framework of two-stage stochastic optimization. This framework formalizes hedging against uncertainty into two stages: in the first, decisions have low cost but the exact input is uncertain; in the second, the input is known but decisions have high cost. Many problems have been cast in this framework, e.g., set cover, minimum spanning tree, Steiner tree, maximum weighted matching, facility location, and knapsack [5, 8, 15, 14]. Prior work has considered linear programming approaches in this framework [23, 25], for example the Sample Average Approximation (SAA) method to reduce the size of a linear program [20, 4]. Ensuring the reduced linear program is representative of the original problem is generally hard and requires problem-specific techniques for most combinatorial optimization settings, however, and so no unified framework has been developed so far.

Our problem is most closely related to bipartite matching problems in this literature. Katriel et al. [18] consider such a problem, where an optimizers wants to buy an edge set containing a maximum matching at the least cost, and must balance fixed first-stage edge costs against the potential risks and rewards of random second-stage edge costs. They propose a polynomial-time deterministic algorithm which approximates the expected cost of minimum matching within a factor of $O(n^2)$, where n is the size of the input graph. They also design a polynomial-time bicriteria randomized algorithm which returns, with probability $1 - e^{-n}$, a matching of size at most $(1 - \beta)n$ which approximates the optimum cost within a factor of $1/\beta$. In our setting, however, we *must* book a room for every buyer served in the first stage, and this bicriteria algorithm gives no guarantees on the set of served but unmatched buyers – they might even all have demanded the exact same option. We seek an algorithm assigning few customers to each option, even in the worst case, an objective that requires significant new insight compared to the setting of [18]. We design an algorithm which assigns at most two customers to each option. Kong and Schaefer [21] give results for the maximum-weighted matching problem, but this objective fails to capture either of our problems.

Maximizing a non-negative submodular function has been extensively studied in the literature (see, e.g., [6, 10, 7, 26]). This problem generalizes the NP-hard max-cut problem [12]. The first constant-factor approximation algorithm for maximizing a non-negative *non-monotone* submodular function was proposed by Feige, Mirrokni, and Vondrak [6]. They present a randomized local-search algorithm with an approximation factor of 0.4. They also show that it is impossible to get a better than 0.5 approximation for the submodular maximization problem with polynomially many oracle queries. Gharan and Vondrak [10] improve this approximation factor to 0.41 by a simulated annealing algorithm. This approximation ratio was further improved to 0.42 by Feldman, Naor, and Schwartz [7] based on a structural continuous greedy algorithm. Later, Buchbinder et al. [3] improved this approximation ratio to the optimal 0.5. It is worth mentioning that submodular maximization plays an important role in many optimization problems, e.g., influence maximization [19, 22], graph cut problems [24], and load balancing [24].

2. PROFIT MAXIMIZATION

Our first step is to consider the second-stage of the coordinator's optimization problem more closely. Note that we let customers form any pack of options they like. Since packs can now intersect in arbitrary ways, the problem of choosing how to assign buyers to options once prices are revealed becomes more complicated. We shall show, however, that it still has nice structure. In this section, we show that the coordinator's objective function has good structural properties. In particular, we show that the profit objective function is submodular. In order to show the submodularity of the profit function, we first prove the expected cost for satisfying a set of buyers $S \subseteq B$ in the second stage is supermodular in S, where the cost of satisfying a set of buyers $S \subseteq B$ is defines as follows:

$$\mathrm{C}(S) = \mathrm{E}[\textstyle\sum_{h \in \mathcal{M}^I(S)} c_h^I],$$

where $\mathcal{M}^I(S)$ is the minimum matching that covers buyers in S in scenario I, and the expectation is over which I occurs. We then leverage the supermodularity of the cost function and prove the profit function is submodular.

We now show that the expected cost for reserving a set of buyers $S \subseteq B$ in the second stage is supermodular in S. We begin by showing that for any fixed future scenario $I \in \mathcal{I}$, the cost $\mathrm{C}_I(S) = \sum_{h \in \mathcal{M}^I(S)} c_h^I$ of reserving options for a set $S \subseteq B$ is supermodular in S. Since our expected cost overall is just a weighted sum of the costs in each possible scenario, it immediately follows that the expected cost of serving a set of buys is supermodular as well. Thus, for the rest of this section, our discussion and arguments fall within the context of a single fixed future scenario $I \in \mathcal{I}$, and so omit it from our notation. Before we begin our proof, however, we first define some notation that will prove useful. First, given a set of buyers S, let $\mathcal{M}(S)$ denote the minimum-cost matching of buyers S to options. Note that after fixing a scenario, multiple matchings may give the same cost; careful tie-breaking is critical to our proofs, and so we defer further discussion of this matter until later. Lastly, we use $\mathrm{C}(S) = \sum_{h \in \mathcal{M}(S)} c_h$ to denote the minimum cost to serve a set of buyers S in our fixed scenario.

We now proceed to show that the function $\mathrm{C}(S)$ is supermodular in S, that is

$$\mathrm{C}(T \cup \{b\}) - \mathrm{C}(T) \geq \mathrm{C}(S \cup \{b\}) - \mathrm{C}(S)$$

for any $S \subseteq T \subseteq B$ and $b \in B \setminus T$. We start by finding a clean characterization of how adding buyers to our served set changes the optimal matching to options. In the following lemma, for any two sets A and B, $A \Delta B$ refers to the symmetric difference of A and B, i.e. an element exists in $A \Delta B$ if and only if, it exists in exactly one of A or B.

LEMMA 2.1. *For any $S \subseteq T \subseteq B$ and any choice of $\mathcal{M}(S)$, there exists a choice of $\mathcal{M}(T)$ such that $\mathcal{M}(S) \triangle \mathcal{M}(T)$ consists of $|T \setminus S|$ disjoint paths of odd length. Furthermore, each of these paths has one endpoint in $T \setminus S$.*

PROOF. Choose $\mathcal{M}(T)$ to be the minimum-cost matching covering T such that the size of $\mathcal{M}(S) \triangle \mathcal{M}(T)$ is minimized. First, note that in $\mathcal{M}(S) \triangle \mathcal{M}(T)$, every element of $T \setminus S$ has degree exactly one; every element of S has degree either zero or two; every other element of B has degree zero; and every element of H has degree zero, one, or two. As such, we can immediately see that $\mathcal{M}(S) \triangle \mathcal{M}(T)$ can be decomposed into a disjoint union of paths and cycles, and the latter must all be of even length since our underlying graph is bipartite. We shortly show that if an even length path or cycle exists, we can use it to modify $\mathcal{M}(T)$ and get a minimum-cost matching that covers T but has strictly smaller symmetric difference with $\mathcal{M}(S)$. The claim immediately follows, since this means $\mathcal{M}(S) \triangle \mathcal{M}(T)$ is a disjoint union of paths of odd length, and as we already observed the set of vertices in B with degree one is precisely $T \setminus S$.

Let $\mathcal{C}$ be any cycle of even length in $\mathcal{M}(S) \triangle \mathcal{M}(T)$. Consider what it represents in the context of our original problem. It means that both of our matchings assigned the customers incident to $\mathcal{C}$ to the options incident to $\mathcal{C}$, just in a different order. Thus, $\mathcal{M}(T) \triangle \mathcal{C}$ would still be a minimum-cost matching, but have strictly smaller symmetric difference with $\mathcal{M}(S)$. Similarly, let $\mathcal{P}$ be an even length path in $\mathcal{M}(S) \triangle \mathcal{M}(T)$. Note that the endpoints of the path must lie in H – otherwise, the set of buyers served by $\mathcal{M}(S)$ and $\mathcal{M}(T)$ would be incomparable, rather than the former being a subset of the latter. Thus, we can see that in the context of our problem, the path $\mathcal{P}$ represents that the two matchings used served the incident buyers using slightly different sets of options. If an option has degree one in $\mathcal{M}(S) \triangle \mathcal{M}(T)$, however, we may conclude that it is used in precisely one of the matchings. Thus, it follows that both $\mathcal{M}(S) \triangle \mathcal{P}$ and $\mathcal{M}(T) \triangle \mathcal{P}$ are valid matchings covering S and T, respectively. Since both $\mathcal{M}(S)$ and $\mathcal{M}(T)$ are minimum-cost matchings, however, we may conclude either of these assignments of the buyers incident to $\mathcal{P}$ to options have the same cost. As such, $\mathcal{M}(T) \triangle \mathcal{P}$ is a minimum-cost matching that has strictly smaller symmetric difference with $\mathcal{M}(S)$, contradicting our choice of $\mathcal{M}(T)$. Thus, we may conclude that no paths of cycle of even length exist in $\mathcal{M}(S) \triangle M(T)$. $\square$

We may use the above lemma to show that the cost function is, in fact, supermodular.

LEMMA 2.2. *For any $S \subseteq T \subseteq B$, and any $b \in B \setminus T$, we have that* $\mathrm{C}(T \cup \{b\}) - \mathrm{C}(T) \geq \mathrm{C}(S \cup \{b\}) - \mathrm{C}(S)$.

PROOF. Consider applying Lemma 2.1 to the sets $T \cup \{b\}$ and S, and some minimum-cost matching $\mathcal{M}(S)$. Let $\mathcal{P}_b$ be the resulting path with endpoint b, and let $\mathcal{P}_{T \setminus S}$ be the union of the paths with endpoints in $T \setminus S$. Observe that each of these paths is an alternating path with respect to $\mathcal{M}(S)$, and that since they are disjoint they can be applied one-by-one to $\mathcal{M}(S)$ in any order to produce a sequence of matchings. Since every path has odd length, we can see that it will increase the size of the matching by one and the cost of the matching by precisely the cost of the option that is one of the path's endpoints. But then, $\mathcal{M}(S) \triangle \mathcal{P}_b$ is a matching covering $S \cup \{b\}$, and so has cost at least $\mathrm{C}(S \cup \{b\})$. Similarly, $\mathcal{M}(S) \triangle \mathcal{P}_{T \setminus S}$ is a matching covering T, and so has cost at least $\mathrm{C}(T)$. But then we can see that

$$\mathrm{C}(T \cup \{b\}) - \mathrm{C}(S) \geq (\mathrm{C}(S \cup \{b\}) - \mathrm{C}(S)) + (\mathrm{C}(T) - \mathrm{C}(S)).$$

Rearranging terms gives precisely the desired inequality. $\square$

THEOREM 2.3. *The profit function is submodular.*

PROOF. We first write the profit objective function as follows:

$$P(S) = \sum_{b\in S} v_b + \mathrm{E}[\sum_{h\in H} c^I_h] - \mathrm{C}(S).$$

Knowing facts that $\mathrm{C}(S)$ is supermodular (based on Lemma 2.2), $\mathrm{E}[\sum_{h\in H} c^I_h]$ is a constant independent of S, and $\sum_{b\in S} v_b$ is just an additive function, we can conclude that the profit function is submodular. □

3. LOSS MINIMIZATION

Unfortunately, the loss objective remains hard to approximate as well. First, in Theorem 3.2 we show that it is not possible to optimally minimize the loss function in polynomial time unless $P = NP$. Moreover, while we can phrase our problem as an integer program, we can show that the integrality gap of this program is quite large. This motivates us to try relaxing some of the constraints in our problem, and find a bicriteria-style approximation. In fact, an optimum solution to the loss minimization problem is an optimum solution to the profit maximization problem as well. Thus, from Theorem 3.2, one may immediately conclude that, there is no polynomial time algorithm for the profit maximization problem unless $P = NP$.

In Theorem 3.2 we used the hardness of 3-*dimensional matching* problem to show that there is no polynomial algorithm for the loss minimization problem unless P=NP.

DEFINITION 3.1. *The* 3*-dimensional matching is defined as follow. Let R_1, R_2 and R_3 be disjoint and finite sets s.t. $|R_1| = |R_2| = |R_3| = n$ and let R be a subset of $R_1 \times R_2 \times R_3$. The problem is to check whether there exists $M \subseteq R$ s.t. $|M| = n$ and for any two distinct triples (r_1, r_2, r_3) and (r'_1, r'_2, r'_3) in M we have $r_1 \neq r'_1$, $r_2 \neq r'_2$ and $r_2 \neq r'_2$.*

The 3-dimensional matching problem is known to be NP-complete [17].

THEOREM 3.2. *There is no polynomial algorithm for the loss minimization problem unless P=NP.*

PROOF. Consider the decision version of our loss minimization problem (DLMin), that we want to know weather the optimum solution is less than k or not. Here, we give a reduction from 3-dimensional matching problem to DLMin. This, in fact, means that DLMin is NP-hard and there is no polynomial algorithm for the loss minimization problem unless $P = NP$.

Let $R \subseteq R_1 \times R_2 \times R_3$ be an instance of 3-dimensional matching problem with $|R_1| = |R_2| = |R_3| = n$ and $R = m$. We create an instance of DLMin with three future scenarios as follow:

- For every item $r = (r_1, r_2, r_3) \in R$ we have one customer with value 1.
- For each element in $R_1 \cup R_2 \cup R_3$ we have an option.
- Each customer corresponds to an item $r = (r_1, r_2, r_3)$ accepts options correspond to elements r_1, r_2 and r_3.
- We have 3 scenarios s.t. in scenario i, the price of options in R_i are 0 (low cost) and price of all other options are 4 (high cost). Each of the scenarios happen with probability $\frac{1}{3}$.

Consider that, since $1 \leq \frac{1}{3} \times 4$, we prefer not to choose a customer rather than matching her to a high cost option even with probability $\frac{1}{3}$. Moreover, in the instance constructed above, each customer in each scenario has exactly one low cost option. This means that, items corresponds to the customers that we select in an optimal solution do not share any element r_i. Therefore, any optimum solution with loss k to the above instance gives a 3-dimensional matching of size $m - k$ in R.

On the other hand, if we choose the customers correspond to the items in a 3-dimensional matching of size $m-k$, we have a solution with loss k to the above instance. This completes the reduction and says that here is no polynomial algorithm for the loss minimization problem unless P=NP. □

$$\begin{aligned}
\text{min:} \quad & \Big(\sum_{b\in B}(1-Y_b)v_b + \frac{1}{N}\sum_{1\leq k\leq N}\sum_{(b,h)\in E} x_{hbk}c_{hk}\Big) \\
\text{s.t.} \quad & \sum_{h\in H} x_{hbk} \geq Y_b \qquad \forall b \in B, \forall 1 \leq k \leq N \qquad (1) \\
& \sum_{b\in B} x_{hbk} \leq 1 \qquad \forall h \in H, \forall 1 \leq k \leq N \qquad (2) \\
& Y_b, x_{hbk} \in \{0,1\} \quad \forall h \in H, \forall b \in B, \forall 1 \leq k \leq N
\end{aligned}$$

Figure 2: The second-stage allocation integer program

We first investigate how the coordinator can minimize the loss objective $L(S)$ via an integer program. One immediate challenge that we encounter is that there may be many possible future scenarios. This is problematic because calculating L requires computing the expected value of the maximum matching in every possible future scenario. We show that we can find an approximate solution $\hat{L}$ to function L using polynomially-many samples of scenarios (See Theorem 4.1) and solving the problem simultaneously for these samples based on a result of [4]. Therefore, our goal is to minimize function $\hat{L}$. The integer program for the problem of minimizing function $\hat{L}$ can be written as in Figure 2. We call it the second-stage allocation IP.

In the second-stage allocation IP: (i) N is the number of samples; (ii) c_{hk} is the known price of option h in the k^{th} sample; (iii) variable Y_b is 1 if and only if $b \in S$, and $Y_b = 0$ otherwise; and (iv) variable x_{hbk} is 1 if and only if option h is assigned to buyer b in the k^{th} sample and is 0 otherwise. Constraint (1) requires that if $Y_b = 1$, then at least one option should be assigned to this buyer in every sample $1 \leq k \leq N$. We call this family of constraints the *capacity* constraints. Constraint (2) for each option $h \in H$ and each scenario k requires that option h in sample k can be assigned to no more than one buyer; we call this family of constraints the *assignment* constraints. The objective function of this IP is exactly equal to $\hat{L}(S)$. Let $\sum_{b\in B}(1 - Y_b)v_b$ be the *lost* term, and $\sum_k \sum_{(b,h)\in E} x_{hbk}c_{hk}$ be the *cost* term of the objective function. In the following, we relax the last two constraints of this IP to their linear counterparts $x_{hbk} \in [0, 1]$ and $Y_b \in [0, 1]$ to obtain a linear program. Edge between buyer b and option h in sample k is a *fractional* edge in a LP solution $\langle \mathbf{x}, \mathbf{Y}\rangle$ if and only if $0 < x_{hbk} < 1$.

The first question that comes to mind when trying to find a minimizer of function $\hat{L}$ is whether we can use a solution to the LP to find an exact or an approximate solution to the IP. However, we show that the integrality gap between the IP and LP solutions can be quite high by the following theorem. Indeed, this theorem shows a linear gap base on the number of buyers, which is a logarithmic gap base on the number of scenarios.

THEOREM 3.3. *The integrality gap of the second-stage allocation IP is at least $\Omega(n)$ where the total number of buyers is $O(n)$.*

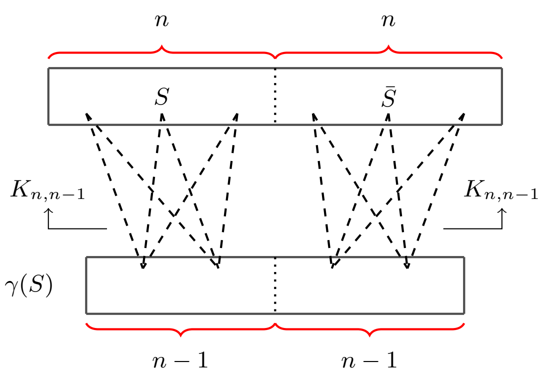

Figure 3: In this graph the upper vertices show the buyers with a subset of size n labeled as S and the rest of buyers labeled as $\bar{S}$. The lower vertices show the group of options $\gamma(S)$. Note that in exactly one scenario all the options in $\gamma(S)$ have a zero price and they have very high prices in all other scenarios. Therefore, we can only match the buyers to options in $\gamma(S)$ in one scenario, and no matching can use this set of options in any other scenario. The best LP solution requires us to assign an amount $\frac{1}{n}$ to each edge in this figure in the aforementioned scenario, and to assign a value equal to zero to all these edges in all other scenarios.

PROOF. Assume there are $2n$ buyers and $(2n-2)\binom{2n}{n}$ options. Suppose there are only two possible prices for each option: the *low* price for all options is 0 and the *high* price for all options is $(2n+1)\binom{2n}{n}$. For each buyer b, $v_b = 1$. Partition the options into $\binom{2n}{n}$ groups of size $2n-2$ each. Consider all the subsets of buyers of size n. There are $\binom{2n}{n}$ such subsets. Let γ denote any one to one mapping from these subsets to option groups. For each subset S and the group of options $\gamma(S)$ mapped to it add edges from each $b \in S$ to all the first $n-1$ options in $\gamma(S)$ and add edges from each $b \notin S$ to all the last $n-1$ options in $\gamma(S)$. Figure 3 illustrates the edges between the buyers and the group of options $\gamma(S)$ for an arbitrary subset of buyers S of size n. Consider $\binom{2n}{n}$ samples and in sample i suppose all the options in the i-th group have the low price (which is 0) and all other options have the high price.

Since the high price for options is very large, we can never assign a person to an option that has a high price. This is true because a trivial solution to the IP is when all the variables are equal to 0 in which case the objective value would be equal to $2n$. However, if we assign a buyer to an option with high price for even one sample, the objective would become at least $2n+1$, which is worse than the trivial solution. Therefore, in any optimal solution to the IP we can assume x_{hbk} is 1 only if option h in sample k has price 0. Thus, the second term in the objective function is always equal to 0, and the objective function for this example simplifies to $\sum_{b\in B}(1-Y_b)$, the number of buyers that we fail to serve.

Note that the objective of the IP in this example is to maximize the number of buyers served. Recall, however, our earlier observation that an optimal integral solution can never use an option with high cost. By construction, this means that an optimal IP solution can only serve at most $n-1$ buyers. Note this property is much less restrictive for LP solutions. Simply serving every buyer at a rate of $(1-1/n)$ will satisfy this requirement while serving a (fractional) total of $2n-2$ customers. We now proceed to formalize the above discussion and demonstrate that it yields the desired gap.

We present a feasible LP solution for which the objective value is equal to 2. Then we show no IP solution can achieve an objective value of less than $n+1$ and we conclude the integrality gap between the LP and IP solutions can be very large.

Here we present a feasible LP solution.

$$Y_b = \frac{n-1}{n} \qquad \forall b \in B \tag{3}$$
$$x_{hbk} = \begin{cases} \frac{1}{n} & \text{if } c_{hk} = 0 \\ 0 & \text{otherwise} \end{cases} \qquad \forall h \in H, \forall b \in B, \forall 1 \le k \le N$$

Consider the first set of constraints in the LP. For each buyer $b \in B$ and each sample, b has edges to $n-1$ options with low prices. Therefore, for $n-1$ options $x_{hbk} = \frac{1}{n}$. This means $\sum_{h\in H} x_{hbk}$ in each sample for each buyer b is $\frac{n-1}{n}$, which is equal to Y_b and the constraint holds. Now consider the second set of constraints. For each option $h \in H$ and each sample, at most n buyers have an edge to this option. The maximum possible value of the variables x is $\frac{1}{n}$. Therefore, $\sum_{b\in B} x_{hbk}$ cannot exceed 1. Thus the set of variables form a feasible LP solution. The objective value with this set of variables would be equal to $\sum_{b\in B}(1-\frac{n-1}{n}) = 2$.

On the other hand, assume an IP solution can achieve an objective value less than $n+1$. Since the objective function is equal to $\sum_{b\in B}(1-Y_b)$, this means for at least n buyers $Y_b = 1$. Consider S to be the set of these n buyers. Consider the sample which has the group of options $\gamma(S)$ as its low price options. In this sample, these n buyers have edges to only $n-1$ options with low prices. It means at least one of them is matched to an option with the high price. Therefore, Y_b can be equal to 1 for at most $n-1$ of them contradicting the fact that $Y_b = 1$ for all the buyers $b \in S$. Therefore, no IP solution can achieve an objective value less than $n+1$.

Therefore, while the optimum LP solution is less than or equal to 2, the optimum IP solution cannot be less than $n+1$. Since, the number of vertices is $2n+(2n-2)\binom{2n}{n} = O(4^n)$, we conclude the integrality between our LP and IP is linear in the total number of buyers and logarithmic in the total number of options. □

The result of Theorem 3.3 leads us to consider relaxations of our problem. In particular, we consider relaxing the constraint that requires matching at most one customer to each option. We will allow ourselves to match **up to two** buyers to an option, but try to minimize the fraction of buyers who are not matched uniquely. We say a buyer is **multi-covered** in a scenario if she is matched to the same option as a previous buyer in that scenario: if we match 2 buyers to an option then one of them is multi-covered. We formally define the bicriteria-style approximation below.

DEFINITION 3.4. *An (α, β)-approximate solution to the second-stage allocation packs IP is a solution which has an objective value at most α times the objective value of the optimal solution to this IP while the number of buyer vertices that it multi-covers in all graphs overall is no more than β times the number of buyer vertices that it covers in all graphs overall.*

THEOREM 3.5. *For any given f such that $0 < f < 1/2$, we can find in deterministic polynomial time, an $(1/f, \min\{\frac{f}{1-2f}, \frac{1}{2}\})$-approximate solution to the second-stage allocation IP in which in every scenario, any option is matched to at most two buyers.*

PROOF. The four-step algorithm supporting Theorem 3.5 is parametrized by $0 < f < 1/2$ and is described next. The primary work done is for Step 4, as seen below.

Step 1: Solving the LP. Solve the LP relaxation; let $x^{(1)}$ and $y^{(1)}$ denote the vectors x and Y of the LP, that occur as the optimal solution-vectors.

Step 2: Filtering. Update $y^{(1)}$ to $y^{(2)}$ as follows: for all b such that $y_b^{(1)} \leq 1-f$, set $y_b^{(2)} := 0$, with $y_b^{(2)} = y_b^{(1)}$ for all other b. Let $x^{(2)} := x^{(1)}$.

Step 3: Scaling up. Update $y^{(2)}$ to $y^{(3)}$ as follows: for all b such that $y_b^{(2)} > 0$, set $y_b^{(3)} := 1$ (we have $y_b^{(3)} = 0$ for all other b). Next update $x^{(2)}$ to $x^{(3)}$ in two sub-steps as follows:

- for all b such that $y_b^{(3)} = 1$, and for all (h,k), set $x_{hbk}^{(3)} := x_{hbk}^{(2)}/y_b^{(1)}$, so that the constraints (1) are satisfied; for all other (h,b,k), initialize $x_{hbk}^{(3)} := x_{hbk}^{(2)}$;
- arbitrarily decrease the $x_{hbk}^{(3)}$ values (subject to non-negativity) such that equality now holds in the constraints (1).

Step 4: Derandomized Dependent Rounding. Separately for each scenario k, we apply a certain derandomized version of the bipartite dependent-rounding procedure of [9] to the vector $x^{(3)}$ (restricted to the index k): the details are as follows. Let $\ell_k(h) = \sum_b x_{hbk}^{(3)}$ denote the fractional load on option h. This procedure rounds x_{hbk} for each (h,b) – recall that we are considering any fixed k now – to some $X_{hbk} \in \{0,1\}$, such that the following properties hold, among others:

(P1) For all (h,b), $E[X_{hbk}] = x_{hbk}^{(3)}$;

(P2) For all b such that $y_b^{(3)} = 1$, $\sum_h X_{hbk} = 1$ with probability one, and

(P3) For all h, $\sum_b X_{hbk} \in \{\lfloor \ell_k(h) \rfloor, \lceil \ell_k(h) \rceil\}$ with probability one.

We will run a derandomized version of this procedure as follows. For $i = 1,2,3$, let L_i and C_i denote the "lost" and "cost" values of the objective function for scenario k, at the end of step i above. That is, for $i = 1,2,3$, at the **end** of Step i above, let

$$L_i = \sum_{b \in B} (1 - y_b^{(i)}) v_b \;\text{ and }\; C_i = \sum_{(b,h)} x_{hbk}^{(i)} c_{hk}.$$

Let $t = \sum_b y^{(3)} b$ be the final number of buyers chosen, and define $H_k = \{h : \ell_k(h) > 1\}$; let $s = |H_k|$. Consider the potential function

$$\Phi = \frac{f}{1-f} \cdot \frac{\sum_{(b,h)} X_{hbk} c_{hk}}{C_3} + \frac{1-2f}{1-f} \cdot \frac{\sum_{h \in H_k} [(\sum_b X_{hbk}) - 1]}{\min\{tf, sf/(1-f)\}}.$$

At every step of the dependent-rounding procedure of [9] – which randomizes among two choices and continually updates the vector X which initially starts at $x^{(3)}$ – deterministically make the choice that never increases Φ. As pointed out in [9], this is indeed possible ((P1) and the linearity of expectation, along with the nature of the choices made in [9], justify this).

Analysis of the algorithm. Let us start with L_i. It is easy to see that $L_2 \leq L_1/f$, and that L_i does not decrease any further. Thus, the "lost" value gets blown up by a factor of at most $1/f$, as compared to the initial LP value.

Note next that $x_{hbk}^{(3)} \leq x_{hbk}^{(1)}/(1-f)$. Combined with (2), this shows that $\ell_k(h) \leq 1/(1-f) \leq 2$ for all (h,k). Thus, property (P3) assures us that the final load $\sum_b X_{hbk}$ on option h in scenario k will be at most two.

To analyze the cost and overbooking, we first claim that for all $h \in H_k$,

$$\sum_h (\ell_k(h) - 1) \leq \min\{tf, sf/(1-f)\}. \tag{4}$$

To see this, start by recalling that $\ell_k(h) \leq 1/(1-f)$ and note that: (i) the LHS of (4) is

$$\sum_h \ell_k(h) \cdot (1 - 1/\ell_k(h)) \leq \sum_b \ell_k(h) \cdot (1 - (1-f)) \leq tf,$$

and (ii) since $\ell_k(h) \leq 1/(1-f)$, the LHS of (4) is at most $s \cdot (1/(1-f) - 1) = sf/(1-f)$. Thus we have (4).

Therefore we see that Φ is initially at most $\frac{f}{1-f} \cdot 1 + \frac{1-2f}{1-f} \cdot = 1$, and thus never exceeds 1. Thus, the final cost value is at most $\sum_k C_3 \cdot (1-f)/f$. However, since $x_{hbk}^{(3)} \leq x_{hbk}^{(1)}/(1-f)$, this implies that the final total cost is at most the LP's cost times $((1-f)/f) \cdot 1/(1-f)$; thus, just like the "lost" function, the "cost" function again gets blown up by a factor of at most f.

Finally for the multi-covering. It is easy to see that the fraction of people multi-covered at the end is at most $U = (1/t) \cdot \sum_{h \in H_k} [(\sum_b X_{hbk}) - 1]$. Since $\Phi \leq 1$ at the end, this implies that

$$U \leq \frac{1-f}{1-2f} \cdot (1/t) \cdot \min\{tf, sf/(1-f)\}. \tag{5}$$

However, property (P3) shows an additional upper-bound on U:

$$U \leq \frac{\min\{t, 2s\} - s}{t} \tag{6}$$

A case analysis of the minimum of these two upper-bounds (e.g., based on whether s/t is at least or at most $1/2$), we get the bound

$$U \leq \min\{\frac{f}{1-2f}, \frac{1}{2}\}$$

as desired. □

4. APPROXIMATE-OPTIMALITY VIA SAMPLING

Charikar et al. consider general 2-stage stochastic models. In these models, an optimizer must make a decision in the first stage which leads to a known cost in the first stage and an unknown cost in the second stage. For our problem, this first stage decision is choosing which customers to serve. In terms of the loss objective, our first stage cost is the values of customers we do not choose to serve, and our second stage cost is buying options for the customers we choose to serve. To that end, we use

$$g(S) = \sum_{b \notin S} v_b \qquad \text{and} \qquad w(S,I) = \sum_{h \in \mathcal{M}^I(S)} c_h^I$$

to denote the first and second stage costs, respectively, of choosing to serve a set of customers $S \subseteq B$ when second stage scenario $I \in \widehat{\mathcal{I}}$ happens. Recall that $\mathcal{M}_I(S)$ is the the minimum-cost matching between customers in S and the options in second stage scenario I. Thus the loss objective for a future scenario I is $L_I(S) = g(S) + w(S,I)$. The goal is to find a first stage decision $S \subseteq B$ so that $E_I(L_I(S)) = g(S) + E_I(w(S,I)) = L(S)$ is minimized. We call such an S a minimizer for the loss objective. Since the space $\mathcal{I}$ might be very large it is hard to solve the problem of minimizing the function L over the full space $\mathcal{I}$. Instead, we define an approximation $\hat{L}$ of L as follows. Given N independent samples of scenarios $I_1, I_2, ..., I_N$ from the space $\mathcal{I}$, we estimate the function L by $\hat{L}(S) = g(S) + \frac{1}{N}\sum_{1 \leq i \leq N} w(S, I_i)$.

In order to apply Charikar et al.'s theorem, we need to prove some properties on the first and second stage costs. These properties are as follows.

1. Both first and second stage costs must always be nonnegative for all first stage decisions and all future scenarios.

2. There must exist a first stage decision for which the first stage cost is zero and the second stage cost is more than that of any other first stage decision for any future scenarios. That is there must exist a first stage decision $S_0 \subseteq B$ for which $g(S_0) = 0$ and $w(S_0, I) \leq w(S, I)$ for all $S \subseteq B$ and all $I \in \widehat{\mathcal{I}}$. We call this S_0 the *null* decision.

3. There must be a bounded inflation factor. That is if S_0 is the null decision from the previous property, then $w(S_0, I) - w(S, I) \leq \eta g(S)$ should hold for all $S \subseteq B$ and a fixed finite real number η. This means the penalty that we have to pay in the second stage because of making a null first stage decision compared to any other first stage decision is no more than a constant factor of the cost of the other first decision.

In our problem, $g(S) = \sum_{b \notin S} v_b$. Since v_b is nonnegative for all customers $b \in B$, $g(S)$ should also be nonnegative for all S. Moreover, $w(S, I)$ is equal to the cost of the matching $\mathcal{M}_I(S)$; since the weights in this matching represent nonnegative option costs, this must be nonnegative as well. Thus, the first property holds. For the second property, we claim B gives the desired null decision for the first stage. Now, $g(B) = \sum_{b \notin B} v_b = 0$, and for a fixed future scenario I, the optimization problem on the future would be matching all the customers, which must be more costly than matching any other subset of customers. The third property holds for our problem with $\eta = \frac{Max^H}{Min^B}$, where Max^H is the maximum possible option price and Min^B is the minimum value of customers. We can see this because

$$\begin{aligned}\sum_{h \in \mathcal{M}_I(B)} c_h^I - \sum_{h \in \mathcal{M}_I(S)} c_h^I &\leq \sum_{b \notin S} Max^H \\ &\leq \eta \sum_{b \notin S} Min^B \\ &\leq \eta \sum_{b \notin S} v_b = \eta g(S).\end{aligned}$$

Thus, we may apply the following theorem, which is a restatement of a theorem from [4], specialized to our setting. Indeed, the number of samples is polynomial in η, $\frac{1}{\epsilon}$, $\log(|\mathcal{I}|)$ and$\log(\frac{1}{\delta})$, and may not be polynomial in the length of the input.

THEOREM 4.1. *Any exact minimizer $\hat{S}$ of function $\hat{L}$ using $\Theta(\eta^2 \frac{1}{\epsilon^4} \log(|\mathcal{I}|) \log(\frac{1}{\delta}))$ samples of scenarios is a $(1 + O(\epsilon))$-approximate minimizer of L with probability $1 - 2\delta$. That is with probability $1 - 2\delta$, the inequality $L(\hat{S}) \leq (1 + O(\epsilon))L(S)$ holds for all $S \subseteq B$.*

Acknowledgments

Supported in part by NSF CAREER award 1053605, NSF grant CCF-1161626, NSF award CNS-1010789, NSF award CCF-1422569, Google Faculty Research award, DARPA/AFOSR grant FA9550-12-1-0423.

REFERENCES

[1] Ajit Agrawal, Philip Klein, and R. Ravi. When trees collide: an approximation algorithm for the generalized Steiner problem on networks. *SIAM J. Comput.*, 24(3):440–456, 1995.

[2] Aaron Archer, MohammadHossein Bateni, MohammadTaghi Hajiaghayi, and Howard Karloff. Improved approximation algorithms for prize-collecting steiner tree and tsp. *SIAM J. Comput.*, 40(2):309–332, 2011.

[3] Niv Buchbinder, Moran Feldman, Joseph Naor, and Roy Schwartz. A tight linear time (1/2)-approximation for unconstrained submodular maximization. In *Foundations of Computer Science (FOCS), 2012 IEEE 53rd Annual Symposium on*, pages 649–658. IEEE, 2012.

[4] Moses Charikar, Chandra Chekuri, and Martin Pál. Sampling bounds for stochastic optimization. In *APPROX/RANDOM*, pages 610–610, 2005.

[5] Kedar Dhamdhere, Vineet Goyal, R. Ravi, and Mohit Singh. How to pay, come what may: Approximation algorithms for demand-robust covering problems. In *FOCS*, pages 367–378, 2005.

[6] Uriel Feige, Vahab S. Mirrokni, and Jan Vondrak. Maximizing non-monotone submodular functions. In *FOCS*, pages 461–471, 2007.

[7] Moran Feldman, Joseph Seffi Naor, and Roy Schwartz. Nonmonotone submodular maximization via a structural continuous greedy algorithm. In *ICALP*, pages 342–353, 2011.

[8] Abraham D. Flaxman, Alan Frieze, and Michael Krivelevich. On the random 2-stage minimum spanning tree. In *SODA*, pages 919–926, 2005.

[9] R. Gandhi, S. Khuller, S. Parthasarathy, and A. Srinivasan. Dependent rounding and its applications to approximation algorithms. *Journal of the ACM*, 53:324–360, 2006.

[10] Shayan Oveis Gharan and Jan Vondrák. Submodular maximization by simulated annealing. In *SODA*, pages 1098–1116, 2011.

[11] Michel X. Goemans and David P. Williamson. A general approximation technique for constrained forest problems. *SIAM J. Comput.*, 24(2):296–317, 1995.

[12] Michel X. Goemans and David P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *J. ACM*, 42(6):1115–1145, 1995.

[13] Guestmob. `http://guestmob.tumblr.com/ourstory`.

[14] Anupam Gupta, Martin Pál, R. Ravi, and Amitabh Sinha. Sampling and cost-sharing: Approximation algorithms for stochastic optimization problems. *SIAM J. Comput.*, 40(5):1361–1401, 2011.

[15] Nicole Immorlica, David Karger, Maria Minkoff, and Vahab S. Mirrokni. On the costs and benefits of procrastination: approximation algorithms for stochastic combinatorial optimization problems. In *SODA*, pages 691–700, 2004.

[16] David S. Johnson, Maria Minkoff, and Steven Phillips. The prize collecting steiner tree problem: theory and practice. In *SODA*, pages 760–769, 2000.

[17] RichardM. Karp. Reducibility among combinatorial problems. In RaymondE. Miller, JamesW. Thatcher, and JeanD. Bohlinger, editors, *Complexity of Computer Computations*, The IBM Research Symposia Series, pages 85–103. Springer US, 1972.

[18] Irit Katriel, Claire Kenyon-Mathieu, and Eli Upfal.

Commitment under uncertainty: Two-stage stochastic matching problems. *Theor. Comput. Sci.*, 408(2):213–223, 2008.

[19] David Kempe, Jon Kleinberg, and Éva Tardos. Maximizing the spread of influence through a social network. In *KDD*, pages 137–146, 2003.

[20] Anton J. Kleywegt, Alexander Shapiro, and Tito Homem-de Mello. The sample average approximation method for stochastic discrete optimization. *SIAM J. on Optim.*, 12(2):479–502, 2002.

[21] Nan Kong and Andrew J. Schaefer. A factor $\frac{1}{2}$approximation algorithm for two-stage stochastic matching problems. *Eur. J. Oper. Res.*, 172:740–746, 2004.

[22] Elchanan Mossel and Sebastien Roch. On the submodularity of influence in social networks. In *STOC*, pages 128–134, 2007.

[23] David B. Shmoys and Chaitanya Swamy. Stochastic optimization is (almost) as easy as deterministic optimization. In *FOCS*, pages 228–237, 2004.

[24] Zoya Svitkina and Lisa Fleischer. Submodular approximation: Sampling-based algorithms and lower bounds. *SIAM Journal on Computing*, 40(6):1715–1737, 2011.

[25] Chaitanya Swamy and David B Shmoys. Sampling-based approximation algorithms for multi-stage stochastic optimization. In *FOCS*, pages 357–366, 2005.

[26] Jan Vondrák. Symmetry and approximability of submodular maximization problems. *SIAM Journal on Computing*, 42(1):265–304, 2013.