CMSC858F: Algorithmic Lower Bounds: Fun with Hardness Proofs Project
A Two Stage Allocation Problem

Hossein Esfandiari
Melika Abolhassani

December 11, 2014

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 if 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.

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.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 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[1]. 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.

example 0.1 *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

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

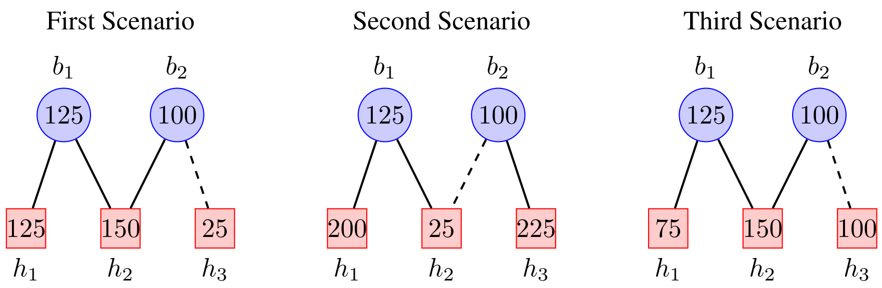

Figure 1: Each graph corresponds to one scenario. The upper vertices show the buyers and the price they are willing to pay. The lower vertices show the options and their realized values in each scenario. The edges indicate the buyers' interest in options. In this example, the best decision is to choose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

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

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

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 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 [3, 6, 10, 9]. Prior work has considered linear programming approaches in this framework [16, 18], for example the Sample Average Approximation (SAA) method to reduce the size of a linear program [13, 2]. 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. [11] 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 [11]. We design an algorithm which assigns at most two customers to each option. Kong and Schaefer [14] 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., [4, 7, 5, 19]). This problem generalizes the NP-hard max-cut problem [8]. The first constant-factor approximation algorithm for maximizing a non-negative *non-monotone* submodular function was proposed by Feige, Mirrokni, and Vondrak [4]. 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 [7] 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 [5] based on a structural continuous greedy algorithm. Later, Buchbinder et al. [1] 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 [12, 15], graph cut problems [17], and load balancing [17].

References

[1] 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.

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

[3] 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.

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

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

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

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

[8] 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.

[9] 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.

[10] 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.

[11] Irit Katriel, Claire Kenyon-Mathieu, and Eli Upfal. Commitment under uncertainty: Two-stage stochastic matching problems. *Theor. Comput. Sci.*, 408(2):213–223, 2008.

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

[13] 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.

[14] 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.

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

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

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

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

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