Algorithmica (2011) 60: 853–876
DOI 10.1007/s00453-009-9378-0

Permutation Betting Markets: Singleton Betting with Extra Information

Mohammad Ghodsi · Hamid Mahini · Vahab S. Mirrokni · Morteza Zadimoghaddam

Received: 9 September 2008 / Accepted: 4 December 2009 / Published online: 16 December 2009

Abstract We study permutation betting markets, introduced by Chen et al. (Proceedings of the ACM Conference on Electronic Commerce, 2007). For these markets, we consider subset bettings in which each trader can bet on a subset of candidates ending up in a subset of positions. We consider the revenue maximization problem for the auctioneer in two main frameworks: the risk-free revenue maximization (studied in Chen et al., Proceedings of the ACM Conference on Electronic Commerce, 2007), and the probabilistic revenue maximization. We also explore the use of some certain knowledge or extra information about the possible outcomes of the market. We first show that finding the optimal revenue in the risk-free model for the subset betting problem is inapproximable. This resolves an open question posed by Chen et al. (Proceedings of the ACM Conference on Electronic Commerce, 2007). In order to identify solvable variants of the problem, we propose the singleton betting language which allows traders to bet an arbitrary value on one candidate for one position. For singleton bettings, we first provide a linear-time implementable necessary and sufficient condition for existence of a solution with positive revenue for any possible outcome. Furthermore, we develop an LP-based polynomial-time algorithm to find the optimum solution of this problem. In addition, we show how to extend this LP-based method to handle some extra information about the possible outcomes. Finally, we consider the revenue maximization problem in a probabilistic setting. For this variant,

M. Ghodsi · H. Mahini (✉) · M. Zadimoghaddam
Department of Computer Engineering, Sharif University of Technology, Azadi St., Tehren, Iran
e-mail: mahini@ce.sharif.edu

M. Ghodsi
e-mail: ghodsi@sharif.edu

M. Zadimoghaddam
e-mail: zadimoghaddam@ce.sharif.edu

V.S. Mirrokni
Microsoft Research, One Microsoft Way, Redmond, WA 98052, USA
e-mail: mirrokni@microsoft.com

we observe that the problem of maximizing the expected revenue is polynomial-time solvable, but we show that maximizing the probability of achieving a pre-specified revenue is $\#P$-Complete.

Keywords Prediction markets · Revenue maximization · Betting markets · Linear programming · Combinatorial algorithms · Matching markets

1 Introduction

Aggregating users' prediction of the outcome of a market has proved very useful in predicting the future. An effective way of extracting users' prediction of the market is observing users' investment on securities. Investment on financial securities such as investment in stock markets is one of the examples of such phenomena. These investments are analogous to *betting* in a financial security. Recently, betting markets have been investigated as a tool to effectively collect the wisdom of the crowd in the market. It has been observed that predictions based on such betting markets are more accurate than other forecasts based on other alternatives such as voting [2–6]. The key reason behind such increase in accuracy is that betting markets incentivize traders to investigate various aspects of the future events, and make precise decisions.

A central problem in betting markets is the problem of matching traders' bets without incurring risk to the auctioneer. The goal in this problem is to find a set of bets such that for any outcome, there is a surplus in the investment of the traders. This problem can be formalized as the revenue maximization problem for the auctioneer in the risk-free setting. Moreover, matching traders' orders as described above helps understanding their use in prediction markets. A set of bets that incurs a risk-free positive revenue for the auctioneer indicates cyclic structures or contradictory bets in these markets. On the other hand, non-existence of such set of bets that guarantees a positive revenue for the auctioneer, may indicate that the users have similar opinions about the outcome of the market, thus it may result in more accurate prediction of the outcome. This analogy shows a relation between the revenue maximization problem for the auctioneer and the quality of the market prediction based on a set of bets. Therefore, the revenue maximization problem in these settings could have applications in evaluating the accuracy of the such predictions.

In this paper, we study the revenue maximization problem in special betting markets known as *permutation markets*, studied first by Chen et al. [1]. In these markets, the outcome is a permutation of a set of candidates V. Traders invest or bet on various securities. The auctioneer collects these bets and either accepts or rejects them. The goal of the auctioneer is to find a subset of bets that incurs a positive surplus, or to find a subset of bets that maximizes his positive surplus, a.k.a. the *revenue*.

1.1 Our Contribution

We study the following two main frameworks in the context of permutation betting markets:

1. *The risk-free setting*: This framework is defined based on the framework of Chen et al. [1], and on the idea of *robust optimization* under uncertainty [7, 9, 10].
2. *The probabilistic setting*: This framework is defined based on the idea of *stochastic multi-stage optimization* [11–13].

In the *risk-free setting*, the goal is to find a subset of traders' bets that guarantees a maximum revenue for the auctioneer in any outcome. This follows the exact setting studied by Chen et al. [1]. This framework is similar to the robust optimization in which the goal is to find a strategy that tends to maximize the objective function in the worst scenario [4, 7, 9, 10]. We study the *subset betting* language in which a trader with a bet $i \in I$ pays $\$b_i$ and bets on one of the two following types of scenarios: (1) one of the candidates ends up in a subset of positions T, or (2) one of the positions is occupied by a subset of candidates S. If the trader's bet is accepted, and the prediction is true, he gets \$1, and nothing if the prediction is false.

Assuming that the surplus money of the traders go to the auctioneer, the revenue maximization problem for the auctioneer is to find a subset of bets that maximizes the revenue of the auctioneer in the worst possible outcome. This problem for subset betting has been posed as a question by Chen et al. [1]. We answer this question by proving that the revenue maximization problem is inapproximable within any factor.

We identify a special case of the problem, called *singleton betting*, that can be solved in polynomial time. In this betting, each trader can bet $\$b_i$ on the security that pays off if a given candidate x_i ends up in a given position y_i. We first provide a necessary and sufficient condition for the existence of a solution with a positive profit in any possible outcome. Next, we present a polynomial-time algorithm for finding the optimal solution of this problem. For this, we first characterize an LP whose optimal integer solution is equal to our optimal solution in the betting problem. Then, we prove that we can change any optimal fractional solution of our LP to an integer solution with the same objective function in polynomial time.

Furthermore, we consider the revenue maximization problem in settings where the auctioneer has some *extra information* about the set of possible outcomes of the market. We show how to use this extra information to find the optimal solution to maximize revenue. It is important to study this extension, since in realistic settings, the auctioneer may have some prior knowledge about the possible outcomes, and he/she should be able to use this information to find a better set of bets to accept. This observation is crucial in some realistic scenarios as it can increase the revenue of the auctioneer by a large amount.

In the *probabilistic setting*, we assume that the auctioneer has a probability distribution over the possible outcomes. In this case, instead of finding a subset of bets that guarantees some revenue, the auctioneer can try to guarantee a revenue of x with high probability. This means that we are willing to take some risk and choose a set of bets that brings us revenue with high probability. This setting follows the idea of stochastic optimization [11–13] in which we have a probability distribution over the possible scenarios, and need to find strategies that optimize in expectation, or with high probability. There are two types of objective functions in these settings. In the first type, the goal is to find a subset of bets that maximizes the expected revenue, given a probability distribution over the possible outcomes. In the second case, the goal is to maximize the probability of achieving at a revenue of at least x (for a given parameter x). We first observe that the problem of maximizing the expected revenue in this model can be solved easily in polynomial time. However, we will show that maximizing the probability of achieving a pre-specified revenue x is $\#P$-Complete.

A preliminary version of this paper appeared in EC'2008 conference [8].

1.2 Related Work

Permutation markets have been introduced by Chen, Fortnow, Nikolova, and Pennock [1]. They study two betting languages for this problem: the pair betting and the subset betting language. In a pair betting language, a trader with bet $i \in I$ pays $\$b_i$ on pairs (a, b) of candidates. If candidate a ends up before candidate b in the outcome, the trader gets \$1, and otherwise he/she gets nothing. The authors consider two types of problems: divisible, and indivisible. In the indivisible problems, the auctioneer can accept or reject each bet. In the divisible version, the auctioneer can accept the bet to an extent y_i where y_i is a real number between 0 to 1. In the divisible setting, the authors show that the problem of maximizing revenue is polynomial-time solvable for subset betting, and is *NP*-complete for pair betting. They also consider the indivisible betting problem, and pose the approximability of the subset betting problem as an open question. After the first draft of this writeup, it was brought to our attention that Conitzer independently found an inapproximability result for this problem [14]. However, his proof is different from ours, and will appear in the journal version of the paper by Chen et al. [1].

Prior to permutation markets, boolean-style markets were studied by Fortnow et al. [15]. In these markets, a possible outcome is one of the 2^n possible 0-1 assignments to a set of n variables. Each trader is allowed to bet on an arbitrary subset of these variables. Traders describe their bets in boolean formulas. The authors show that the matching problem in this setting is *co-NP*-Complete for the divisible variant and $\sum_2^P$-Complete for the indivisible variant.

Another related work to betting markets is the *market scoring rule* mechanism defined in [16]. In this setting, a joint probability distribution across all outcomes is given, and traders bet on a combinatorial number of outcomes. One main difference between this setting and the framework considered in this paper is that the traders arrive sequentially, and the market maker pays to the last trader. In this setting, he may incur some loss. This is similar to our probabilistic setting in which the auctioneer may also incur loss with some probability. However, in the risk-free setting, when the trader accepts some bets, he/she does not bear any risk.

The risk-free setting considered in this paper is related to robust combinatorial optimization [7, 9, 10] in which given a set of possible scenarios that can happen in the future, the goal is to find a strategy that optimizes the objective function in the worst scenario. A challenging and interesting aspect of permutation betting markets is that the number of possible outcomes is $n!$ which is exponential on the size of the input. This is similar to the robust optimization framework with exponential number of scenarios. It has been proved that such robust optimization problems with exponential number of scenarios are harder to approximate [10]. The probabilistic setting considered in this paper is similar to stochastic optimization [11–13] in which given a probability distribution over the possible scenarios that can happen in future, the goal is to find a strategy that optimizes the expected objective function. However, the literatures on both robust and stochastic optimization consider combinatorial optimization problems like network design and covering problems, and not the permutation problems considered in this paper.

The problem of allocating items to bidders in combinatorial auctions to maximize the auctioneer's revenue is considered in [17–21]. However, in contrast to our settings, the risk and uncertainty concepts are not considered in most of these works.

1.3 Organization

This paper is organized as follows. First, in Sect. 2, we formally define permutation markets and the subset betting problem. In Sect. 3, we define the subset betting problem, and prove its inapproximability. In Sect. 4, we first formally define the singleton betting problem, and give an algorithm to verify if there exists a subset of bets with positive revenue. Then, we provide a linear programming-based polynomial-time algorithm to maximize the revenue for the singleton betting problem. At the end of Sect. 4, we show how to solve the revenue maximization problem in the presence of some extra information about the possible outcomes of market. Finally, in Sect. 5, we define the probabilistic setting and present a positive remark and a negative result for this setting.

2 Preliminaries

In this section, we formally define permutation betting markets and the subset betting problem.

Permutation Betting Markets Permutation betting markets are those in which the set of possible outcomes of the market is the set of all possible permutations of n candidates. For example, the candidates can be horses in a race, and the outcome is the ranking of horses in an increasing order. In such markets, traders can bet on various types of securities for a future event. The result of the future event determines the outcome of the market. For example, the event could be a horse competition. In permutation markets, each security is a property of the ranking outcome. The value of the security is not known before the event, and its truth will be revealed after the future event. For example, a security is "horse A ends up in position 3." The auctioneer receives a set of bets on various types of securities, and can accept or reject each bet. Each bet i consists of a bet value b and a security ϕ. b is the amount of money the trader is willing to pay if his/her bet is accepted. If the bet is accepted by the auctioneer, the trader pays \$$b$ before the event, and after the event, if the security ϕ happens, e.g., if horse A ends up in position 3, then, the trader gets \$1. The revenue of the auctioneer is defined as follows. If the auctioneer accepts a bet of value b on a security ϕ, if ϕ happens, the auctioneer's revenue from this bet is $b - 1$, and if ϕ does not happen, the revenue from this bet is b. The (total) revenue of the auctioneer is the sum of his/her revenue from all accepted bets. In the risk-free setting, the goal of the auctioneer is to find a subset of bets that guarantees a positive revenue for him/her in any possible outcome. For example, if one trader bets on the event "horse A ends up in position 2" for \$0.7 (i.e., the trader pays \$0.7 ahead of time and gets \$1 if the event happens), and another trader bets on the event "horse B ends up in position 2" with \$0.7, then the set of all two bets is a risk-free set of bets for the auctioneer, since by accepting the two bets, in any possible outcome, the auctioneer has to pay \$1 to at most one trader, and thus the revenue of the auctioneer is $2 \times 0.7 - 1 = 0.4$. Our goal is to find a subset of bets for the auctioneer to accept in order to maximize the revenue.

Subset Betting A subset betting permutation market allows two types of bets. Traders can either bet on a subset of positions a candidate may end up with, or they can bet on a subset of candidates that will occupy a particular position. In an instance of the subset betting problem, we are given a set of bets, I. A bet $i \in I$ of the first type is a triple (b_i, x_i, Y_i) where b_i is the amount of money that the trader is willing to pay, x_i is the candidate he is bidding on, and Y_i is a subset of positions. The trader gets \$0 if candidate x_i does not end up in a position in set Y_i, and gets \$1 when candidate x_i stands at one of the positions in set Y_i. A bet $j \in I$ of the second type is a triple (b_j, X_j, y_j) where b_j is the amount of money that the trader is willing to pay, X_j is the set of candidates he is bidding on, and y_j is a position. The trader gets \$0 if none of the candidates in X_j ends up in position y_j, and gets \$1 if one of the candidates in set X_j stands at position y_j.

3 Hardness of Subset Betting

In this section, we show that it is *NP*-hard to approximate the optimal revenue for subset bettings within any factor.

We say that an algorithm for the revenue maximization problem is a c-approximation algorithm, if for any input market with optimal revenue x, this algorithm runs in polynomial time and returns a solution with revenue not less than cx. We prove that the problem of maximizing revenue in subset bettings is not approximable within any multiplicative factor c even in the special case that all bets are of second type, and y_j is equal to 1 for all bets. We do so by proving that in some instances of the problem, we can not even decide whether or not the optimal answer has positive profit for the auctioneer in every possible outcome. This fact implies that this problem can not be approximated. To see this, assume that the problem admits a c-approximation algorithm. Therefore, using this algorithm, we can verify whether x is zero or a positive number. It remains to prove that verifying whether or not the revenue is positive is *NP*-hard.

Definition 1 In the big independent set problem, we are given a graph G, with n vertices, with no isolated vertex (a vertex of degree zero), and a number $k > n/2$. The goal is to output 'Yes' when the graph has an independent set of size k, and 'No' otherwise.

Lemma 1 *The big independent set problem is NP-Complete.*

Proof We prove the lemma using the *NP*-Completeness of the independent set problem. In the independent set problem, given a graph G and an arbitrary input k, the goal is to find an independent set of size k in G. Without loss of generality, we assume that G has no isolated vertex. Now, we give a reduction from the independent set problem to the big independent set problem. Let the number of vertices and edges in G be n and e, respectively. Let x be the size of the biggest independent set in G. We add n patterns of P_3 (a path with 3 vertices) to G. This new graph has $n+3n=4n$ vertices and $e+2n$ edges. Obviously, the size of the biggest independent set in this new graph is $2n+x$ which is greater than $4n/2$. So the problem of finding the size of the biggest independent set in this new graph reduces to the big independent set problem. Thus, if there is a polynomial-time algorithm which solves the big independent set problem, we can find $2n+x$ using this algorithm, and consequently x can be found. As a result, the big independent set problem is *NP*-Complete. □

Now, we can prove the main theorem of this section which resolves the open question posed by Chen et al. [1].

Theorem 1 *For any $c>0$, there is no c-approximation algorithm for subset betting problem with indivisible bets.*

Proof Let (G,k) be an instance of the big independent set problem. We construct an instance of the subset betting problem as follows. For each edge between vertices u and v of G, we consider a candidate $c_{u,v}$ in our instance. Note that $c_{u,v}=c_{v,u}$. Let ε be a positive number which satisfies inequalities $(k-1)\varepsilon<1<k\varepsilon$ and $n\varepsilon<2$. According to the fact that $k>n/2$, we know that such an ε exists. Now, for each vertex v in G, we insert a trader in our instance of the subset betting problem. This trader's bet is a triple of form $(\varepsilon, X_v, 1)$ where $X_v=\{c_{v,u}|(v,u)\in E(G)\}$.

Now, we show that in order to have an output with a positive profit for any possible outcome, the auctioneer should not accept bets of two traders u and v which are adjacent in graph G. If the auctioneer accepts the bet of both traders u and v, if a candidate $c_{u,v}$ stands at the first position (which is a possible outcome), he/she should pay \$1 to each of these two traders. Thus, in this case, he/she should pay \$2, but all the money that is given to the auctioneer is at most $n\times\varepsilon$ which is less than 2. Note that there are n traders, and each of them pays the auctioneer ε. This implies that if the auctioneer accepts the bets of two incident traders, there is a possible outcome in which his/her revenue is negative. This fact shows that the auctioneer should not accept bets of two adjacent traders u and v. Therefore, the traders whose bets are accepted should form an independent set in G. There are possible outcomes in which we should pay one dollar. So, in order to have a positive revenue, we should accept at least $1/\varepsilon$ number of bets. In other words, we should accept at least $k=\frac{1}{\varepsilon}$ bets which form an independent set of size at least k in G. Therefore, this instance of the subset betting problem has a solution in which the auctioneer's revenue is always positive, the graph has an independent set of size k, and vice versa. Using the previous lemma, we know that this problem is *NP*-Complete. This fact proves that verifying if the revenue is positive or not cannot be done in polynomial time unless $P=NP$. Therefore, for any $c>0$, there is no c-approximation algorithm for the subset betting problem. □

4 The Singleton Betting Problem

In this section, we first formally define the singleton betting problem, and then give a linear-time algorithm for verifying if the auctioneer's revenue is positive. Next, we show that the problem of maximizing revenue for singleton betting can be solved via a linear programming formulation. Finally, we show that this polynomial-time algorithm can be used to solve the same revenue maximization problem in the presence of some extra information about the outcome of the market.

4.1 Definitions and Notations

The Singleton Betting Problem The singleton betting market problem is a special case of the subset betting problem in which players can bet on a singleton set of candidates for a single position. More formally, a singleton betting market allows traders to bet on a (single) position that a (single) candidate may end up with. Consider a set of candidates in a permutation market in which all $n!$ permutations are possible outcomes, where n is the number of candidates. In an instance of the singleton betting problem, we are given a set I of bets that are submitted to the auctioneer by a set of traders. Each bet $i \in I$ is a triple (b_i, x_i, y_i), where x_i is a candidate, y_i is a position, and b_i is the amount which the trader i is willing to pay for a unit share. Similar to the subset betting problem, if bet i is accepted, trader i pays $\$b_i$ before the outcome is revealed, and if candidate x_i stands at position y_i in the outcome, trader i wins \$1 and wins \$0 otherwise. Given a set I of bets, the auctioneer can accept or reject each of the bets. The goal of the auctioneer is to find a subset of bets that maximizes its revenue.

To achieve this goal, we consider the following two problems: *existence problem*, and *revenue maximization problem*. In the existence problem, the auctioneer's goal is to find a subset of bets, called a risk-free subset, such that by accepting this subset, the auctioneer has a positive profit in any possible outcome. In the revenue maximization problem, the auctioneer's goal is to find a subset of bets such that accepting it, the auctioneer maximizes his/her minimum profit in every possible outcome. It is clear that the existence problem is a special case of the revenue maximization problem. There is a generalization in which any trader is allowed to order more than one share of security in her bet. In this case, the auctioneer is allowed to accept any subset of them. We can easily generalize our results to solve this problem.

In the following, we give a simple combinatorial algorithm for the existence problem, and an LP-based algorithm for the revenue maximization problem. First, we define some notations that will be used throughout this section.

Corresponding Bipartite Graph G_I Given an instance of the singleton betting problem with a set of bets I, we construct a bipartite graph G_I as follows. For every candidate, we place a vertex in the upper part of G_I and for every position, we place a vertex in the lower part of G_I. Let U^G denote the set of vertices in the upper part and L^G denote the set of vertices in the lower part. We denote the ith vertex of the upper part by u_i and the jth vertex of the lower part by l_j. Finally, for every triple $(b_i, x_i, y_i) \in I$, we put an edge between $u_{x_i} \in U^G$ and $l_{y_i} \in L^G$ with weight b_i. Note that it is possible to have multiple edges between two nodes in G_I.

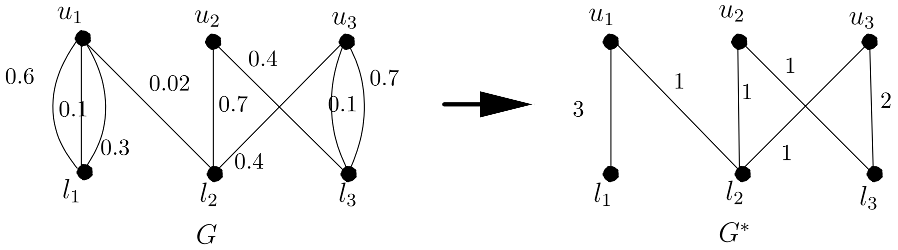

Fig. 1 Definition of G^*

Given a simple edge-weighted bipartite graph $G(U^G, L^G, E)$, let $w^G_{i,j}$ be the weight of the edge between vertices $u_i \in U$ and $l_j \in L$.

Also, given a multigraph $G(V, E)$, let $G^*(V^*, E^*)$ with edge weights w^* be a simple edge-weighted graph with the same set of vertices, i.e., $V^* = V$ such that w^*_{ij}, the weight of the edge between vertices $i, j \in V^*$ in G^*, is equal to the number of edges between vertices i and j in G (as shown in Fig. 1).

Note that $w^{G^*}_{i,j}$ is equal to the number of edges between vertices i and j in G. Each bet has a corresponding edge in G_I. Therefore, when the auctioneer accepts a bet, we can also say that the auctioneer *accepts* the corresponding edge.

For example, consider the singleton betting market depicted in Fig. 1. There are nine bets corresponding to edges of the bipartite graph. As a result, the set of bets, I contains the following triples:

$$(0.6, 1, 1), (0.1, 1, 1), (0.3, 1, 1), (0.02, 1, 2), \ldots, (0.1, 3, 3), (0.7, 3, 3)$$

In this example, if the auctioneer accepts all bets, he/she gets \$3.32 before the outcome. If candidates 1, 2 and 3 stand in positions 1, 2 and 3 respectively, the auctioneers should pay $3 + 1 + 2 = \$6$ to traders.

Accepted Graph If the auctioneer accepts a subset of bets, there is a subgraph in G which is formed by the edges corresponding to the accepted bets. We call this an *accepted graph*. We say an auctioneer will *win* with respect to an accepted graph H, if accepting the bets corresponding to the edges of H gives a positive revenue to the auctioneer in every possible outcome.

Graph Theoretic Preliminaries In every graph G, let M_G be the value of the maximum weighted matching. Vector $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_{|V(G)|})$ is called a *weighted vertex cover* of graph G if for every edge $e = (i, j)$ we have $w^G_e \le \alpha_i + \alpha_j$. The *minimum weighted vertex cover* is a weighted vertex cover which minimizes the sum $\sum_{i \in V(G)} \alpha_i$.

4.2 The Existence Problem

In this section, we prove a necessary and sufficient condition for the existence problem which can be checked in linear time.

Theorem 2 *Given a set of bets I for the singleton betting problem, there exists a set of risk-free bets for the existence problem if and only if there exists a vertex $i \in V(G_I)$ and a set of edges A such that* (i) *edges in A are adjacent to vertex i, and each pair of edges in A has only one endpoint in common which is i; and* (ii) *the total weight of edges in A exceeds 1, or equivalently $\sum_{e\in A} w_e^{G_I} > 1$.*

Proof *Proof of sufficiency*: Suppose that there is a vertex i with the desired properties. Without loss of generality, suppose i is in the upper part of G_I. Assume that there exists a set A of edges which are adjacent to i and the sum of their weights is greater than 1. If the auctioneer accepts the bets corresponding to edges in A and rejects the other ones, the amount of money which is earned by him/her is greater than \$1. On the other hand, the auctioneer must pay at most \$1 in the worst case, because all the accepted bets have the candidate i in common and their positions (the third element in the triples of bets) are distinct, so in each outcome the candidate i stands at only one position, and we lose at most \$1.

Proof of necessity: Suppose there is no vertex i with the desired property of the theorem and there is a subgraph H of G_I such that the auctioneer will win if he accepts the bets corresponding to edges of H. First, we find the subgraph $\hat{H}$ of G_I such that the auctioneer will win if he accepts the corresponding bets of $\hat{H}$ and we have $w_{i,j}^{\hat{H}^*} \le 1$, for all i, j. This means that there exists at most one edge between every pair of vertices $u_i \in U^{\hat{H}}$ and $l_j \in L^{\hat{H}}$ in $\hat{H}$. Finally, we prove that if a subgraph like $\hat{H}$ exists, we will reach a contradiction.

In order to prove the existence of $\hat{H}$, we need the following lemma.

Lemma 2 *Let G^* be a weighted simple bipartite graph with integer weights. If the value of the maximum weighted matching in G^* is $M_{G^*} \ge 1$, then there exists a vertex i with the following property*:

- *If we decrease the weights of all edges adjacent to i by 1 unit, the value of the maximum weighted matching in the remaining graph will be $M_{G^*} - 1$.*

Proof The dual of the maximum weighted matching problem in a bipartite graph G^* is the following problem: assign values α_i and β_j to the vertices of G^* (α_i to vertex $u_i \in U$ and β_j to vertex $l_j \in L$) such that for every edge $e = (u_i, l_j)$, we have that $\alpha_i + \beta_j \ge w_{i,j}$, and we also want to minimize the objective function $\sum_{u_i\in U} \alpha_i + \sum_{l_j\in L} \beta_j$. Based on the weak duality theorem, we know that the minimum feasible value of $\sum_{u_i\in U} \alpha_i + \sum_{l_j\in L} \beta_j$ is greater than or equal to M_{G^*} in G^*. Consider the optimal dual solution α_i and β_j for $u_i \in U$ and $l_i \in L$. Note that the weights of the edges in G^* are integer, therefore, there exists an optimal solution in which all values of α_i and β_j are integers. This is true, since the dual of the weighted matching is totally unimodular, and its integrality gap is 1 [22]. We also know that $M_{G^*} \ge 1$, so at least one α_i or one β_j is greater than 0. Without loss of generality, suppose $\alpha_k > 0$, and because α_k is an integer number, we conclude that $\alpha_k \ge 1$. Now, we can decrease the weights of the edges adjacent to u_k by 1 unit and let G' be the remaining graph. It is clear that

$$(\beta'_j = \beta_j \ \forall l_j \in L,\ \alpha'_i = \alpha_i \ \forall u_i \in U, u_i \ne u_k, \text{and } \alpha'_k = \alpha_k - 1)$$

is a feasible solution for the dual problem in graph G' with value $\sum_{u_i \in U} \alpha'_i + \sum_{l_j \in L} \beta'_j = M_{G^*} - 1$. Therefore, the value of every weighted matching in G' is not greater than $M_{G^*} - 1$. On the other hand, consider the maximum weighted matching in G^* with value M_{G^*}. It is clear that the value of this matching in G' is $M_{G^*} - 1$. So the value of the maximum weighted matching in G' is $M_{G^*} - 1$. □

Now we return to the proof of Theorem 2. Consider a graph H and assume that the auctioneer accepts the bets corresponding to the edges of H. (Note that we assume that the auctioneer will win by accepting these bets.) Let the value of the maximum weighted matching in H^* be M_{H^*}. It is clear that for some permutations, the auctioneer must pay M_{H^*} to the traders. On the other hand, the auctioneer gets $\sum_{e \in H} w_e^H$ amount of money from traders at first. Since the auctioneer is seeking a risk-free subset, we should have:

$$M_{H^*} < \sum_{e \in H} w_e^H \tag{1}$$

If H^* has an edge with weight greater than 1, there are at least two edges in H between the endpoints of that edge with weight greater than 1. We repeat the following procedure iteratively, until there is no edge with weight greater than 1 in H^*.

– We know that there exists a vertex i in H^* with the desired property of Lemma 2. For every vertex j, remove one of the edges between vertices i and j in H. Let $\tilde{H}$ be the remaining graph. According to Lemma 2, if we decrease the weights of edges adjacent to i in H^* by 1 unit, the value of the maximum weighted matching in H^* will decrease by exactly 1 unit. Therefore, we have $M_{\tilde{H}^*} = M_{H*} - 1$. We assume that there is no vertex with the desired property of Theorem 2. Therefore, the sum of the weights of the removed edges from H is not greater than 1, and we have $\sum_{e \in \tilde{H}} w_e^{\tilde{H}} \geq \sum_{e \in H} w_e^H - 1$. Using (1) we conclude:

$$M_{\tilde{H}^*} = M_{H*} - 1 < \sum_{e \in H} w_e^H - 1 \leq \sum_{e \in \tilde{H}} w_e^{\tilde{H}} \tag{2}$$

This proves that if the auctioneer accepts the bets corresponding to the edges of $\tilde{H}$, he wins. Therefore, we can replace H by $\tilde{H}$ and iteratively repeat this procedure until we reach a graph H^* with no edge weight greater than one.

This proves that there exists a graph H such that the auctioneer wins with respect to H, and there is at most one edge between any pair of vertices in H.

Now, we construct a network flow F using the graph H as follows. See Fig. 2.

1. Add two vertices s and t to H. Let s be the source of our network flow and t be its sink.
2. Put an edge between s and each vertex u_i in the upper part of H with capacity of 1.
3. Put an edge between each vertex l_j in the lower part of H and t with capacity of 1.
4. Let the capacity of each edge from vertex u_i in the upper part to vertex l_j in the lower part be equal to w_{u_i,l_j}^H.

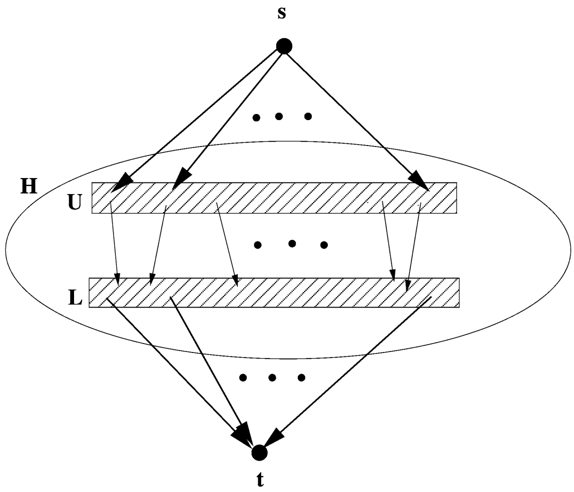

Fig. 2 Constructing the network flow F using graph H

We know that there is no vertex with the desired property of Theorem 2. Therefore, for every vertex i, the sum of the weights of the edges adjacent to i in H is not greater than 1. So, we have a flow with value $\sum_{e\in H} w_e^H$ in the network flow F. Construct a new network flow F' by rounding up the capacities of edges in F. It is clear that the value of the maximum flow in F' is not less than the value of the maximum flow in F, because we do not decrease the capacities. On the other hand, we can see that the value of the maximum flow in F' is equal to the value of the maximum weighted matching in $H^*(M_{H^*})$. Knowing these facts, we can conclude:

$$M_{H^*} = \text{max flow in } F' \geq \text{max flow in } F \geq \sum_{e\in H} w_e^H \tag{3}$$

which is a contradiction (see (1)) □

Verifying the necessary and sufficient condition of Theorem 2 for all vertices in graph G_I can be done in running time $O(|I| + m + n)$ where n and m are the number of candidates and positions respectively. As a result, Theorem 2 gives a linear-time algorithm for the existence problem.

Now we are ready to solve the generalization in which traders can order more than one share of security. In the existence problem, we only need to compute the sum of weights of edges incident to a specific vertex u. Note that if we submit C copies of a bet, these are C parallel edges in G_I with 2 common vertices. Thus, we should consider only one of them in our calculations. Therefore, this generalization is not computationally harder.

4.3 The Revenue Maximization Problem

In this section, we propose a polynomial-time algorithm for finding a subset of bets with the maximum guaranteed revenue to the auctioneer. The algorithm is based on a linear programming (LP) formulation. We first characterize an LP whose optimal

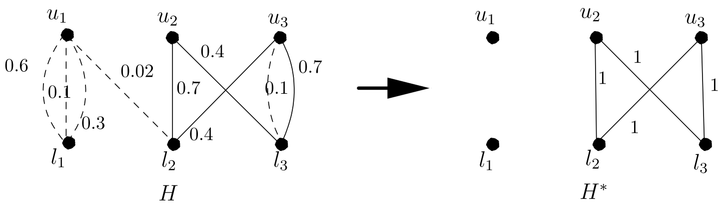

Fig. 3 A solution with revenue 0.2

integer solution is equal to our optimal solution in the singleton betting problem. We relax the linear program to a fractional linear program, and then prove that we can change any optimal fractional solution of our LP to an integer solution with the same objective function in polynomial-time. Note that in the revenue maximization problem, we have a weighted bipartite graph G with multiple edges each of which corresponds to one bet, and we want to find a subgraph H of G that maximizes:

$$\sum_{e\in H} w_e^H - M_{H^*} \tag{4}$$

In other words, if the auctioneer accepts the bets of the edges of graph H, he earns $\sum_{e\in H} w_e^H$ amount of money from traders, and he should pay M_{H^*} in the worst case outcome. For example, consider the singleton betting market shown in Fig. 1. If the auctioneer accepts bets $(0.7, 2, 2)$, $(0.4, 2, 3)$, $(0.4, 3, 2)$ and $(0.7, 3, 3)$, the accepted graph is H that is shown in Fig. 3. Note that $\sum_{e\in H} w_e^H$ is equal to 2.2, and the maximum weighted matching in graph H^*, M_{H^*} is 2. So the revenue is $2.2 - 2 = 0.2$ in this case.

Now we characterize the structure of the maximum revenue solution. Let the subgraph $H \subseteq G$ be the maximum revenue solution. Consider the minimum weighted vertex cover of H^* (see the definition of G^* in the preliminaries). Assume that the value α_i is assigned to the vertex $u_i \in U^{H^*}$, and β_j is assigned to the vertex $l_j \in L^{H^*}$ in this weighted vertex cover.

Lemma 3 *In the optimal accepted graph H, $w_{i,j}^{H^*}$ is equal to* $\min(w_{i,j}^{G^*}, \alpha_i + \beta_j)$ *for every pair of vertices i and j.*

Proof We know that H is a subgraph of G, thus $w_{i,j}^{H^*}$ is not greater than $w_{i,j}^{G^*}$. On the other hand, the values $\alpha_1, \alpha_2, \ldots, \alpha_{|V(H)|}$ and $\beta_1, \beta_2, \ldots, \beta_{|V(H)|}$ form a weighted vertex cover of graph H^*, so we have that $w_{i,j}^{H^*} \le \alpha_i + \beta_j$.

For the sake of contradiction, assume $w_{i,j}^{H^*}$ is not equal to $\min(w_{i,j}^{G^*}, \alpha_i + \beta_j)$ for some i and j, thus $w_{i,j}^{H^*} < w_{i,j}^{G^*}$ and $w_{i,j}^{H^*} < \alpha_i + \beta_j$. The value $w_{i,j}^{H^*}$ is integer, for any pair of vertices i and j. Therefore, by definition, α_i and β_j are also integers, for every i and j. Thus, $w_{i,j}^{H^*} + 1 \le \alpha_i + \beta_j$. We can add one of the edges between i and j in $G - H$ to H. Note that such an edge exists because we assume that $w_{i,j}^{H^*} < w_{i,j}^{G^*}$.

It implies that the value of $w_{i,j}^{H^*}$ increases by one. But, it is clear that the minimum weighted vertex cover of H^* is still a weighted vertex cover. Using the fact that the value of the maximum weighted matching is equal to the value of the minimum weighted vertex cover, we conclude that the value of $\sum_{e\in H} w_e^H - M_{H^*}$ will increase by adding one of edges between vertices i and j to the optimal accepted graph H. This contradicts the optimality of H. □

Lemma 3 implies that using values of α_i and β_j, we can determine the value $w_{i,j}^{H^*}$ by setting it to $\min(w_{i,j}^{G^*}, \alpha_i + \beta_j)$. Now, we can write an integer linear program whose optimal solution determines our optimal solution for the singleton betting problem. In this ILP, we want to find the values of α_i and β_j. We also want to choose edges which should be added to the optimal subgraph H. For ease of notation, let $w_{i,j,t}^G$ be the weight of the tth edge between vertices i and j in G. Without loss of generality assume that the edges between vertices i and j are sorted in decreasing order with respect to their weight such that we have $w_{i,j,t}^{G^*} \geq w_{i,j,t+1}^{G^*}$. The following program is the ILP which helps us in computing the minimum weighted vertex cover:

$$\max\left(\sum w_{i,j,t}^G y_{i,j,t} - \sum x_i - \sum x'_j\right)$$

$$\begin{aligned}
&\sum_{t=1}^{w_{i,j}^{G^*}} y_{i,j,t} = Y_{i,j} \quad \forall i \in U, j \in L \\
&Y_{i,j} \leq x_i + x'_j \quad \forall i \in U, j \in L \\
&x_i, x'_j \geq 0 \quad \forall i \in U, j \in L \\
&y_{i,j,t} \in \{0,1\} \quad \forall i \in U, j \in L, 1 \leq t \leq k_{ij}
\end{aligned} \tag{5}$$

The ILP variables x_i, x'_j, $y_{i,j,t}$ and $Y_{i,j}$ are defined as follows:

- x_i is the value of α_i in the minimum weighted vertex cover of H^*.
- x'_j is the value of β_j in the minimum weighted vertex cover of H^*.
- $Y_{i,j}$ is the value of $w_{i,j}^{H^*}$ which is equal to number of edges between vertices i and j in H.
- $y_{i,j,t}$ is a number which is equal to 0 or 1, and indicates whether the tth edge between i and j in G belongs to H.

By strong duality, the value of the maximum weighted matching is equal to the value of the minimum weighted vertex cover. Since the ILP variables x and x' correspond to the vectors of the minimum weighted vertex cover (α and β), the value $\sum x_i + \sum x'_j$ in the objective function of the ILP is equal to M_{H^*}. As a result, the optimal solution of the integer linear program 5 characterizes the maximum revenue of the singleton betting problem. In order to solve the integer linear program 5, we relax the integer constraints $y_{i,j,t} \in \{0,1\}$ to linear fractional constraints $0 \le y_{i,j,t} \le 1$. As a result, we get the following linear programming relaxation:

$$\begin{aligned}
&\max\left(\sum w^G_{i,j,t} y_{i,j,t} - \sum x_i - \sum x'_j\right)\\
&\sum_{t=1}^{w^{G^*}_{i,j}} y_{i,j,t} = Y_{i,j} \quad \forall i \in U, j \in L\\
&Y_{i,j} \le x_i + x'_j \quad \forall i \in U, j \in L \qquad\qquad (6)\\
&x_i, x'_j \ge 0 \quad \forall i \in U, j \in L\\
&0 \le y_{i,j,t} \le 1 \quad \forall i \in U, j \in L, 1 \le t \le k_{ij}
\end{aligned}$$

We can solve the linear program 6. Now, the question is how to round the solution of 6 and construct an integer solution for program 5. The following Lemma 4 shows that the integrality gap of this linear program is 1 and any solution of this LP can be rounded to an integer solution in polynomial time without changing the value of the objective function. Here, we prove this fact by showing that LP 6 is totally unimodular. For completeness, we give an explicit polynomial-time rounding method for rounding fractional solutions of this LP to optimal integer solutions in the appendix.

Lemma 4 *The integrality gap of LP* 6 *is* 1 *and an optimal integer solution of ILP* 5 *can be found in polynomial-time by solving the LP relaxation* 6.

Proof Here, we prove this lemma by showing that the LP is totally unimodular. For a constructive proof of this lemma, see the Appendix.

There are four types of variables in LP 6, i.e. $y_{i,j,t}, Y_{i,j}, x_i, x'_j$. Let v be a vector that contains all types of variables. We can write the constraints of LP 6 as a matrix inequality $Av \le b$ where A and b are defined as follows: A is a matrix whose number of rows and columns are equal to the number of constraints and variables in the LP respectively, and entries of A correspond to coefficients in the linear constraints of this LP. The vector b contains the right hand side values of the constraints.

There are some equality constraints in LP 6. We can use some slack variables, and replace these equalities with some inequalities, so the constraints can be written in the inequality form $Av \le b$. By the way, these slack variables do not disturb the totally unimodularity property of this inequality system.

It is well known that in order to prove that the integrality gap of LP 6 is 1, it suffices to show that A is totally unimodular [23, 24]. This fact also implies a polynomial-time algorithm for rounding any fractional solution of LP 6 to an optimal integer solution to ILP 5.

For contradiction, assume A is not totally unimodular. In that case, A should have a square submatrix with determinant not equal to 0, 1 or -1. Suppose K is the smallest square submatrix of A with this property.

It is not hard to see that each row or column of K has at least two non-zero entries. Since, if there is a row (or column) with only one non-zero entry a (a is either 1 or -1), we can say that the absolute value of determinant of K is equal to the absolute value of determinant of K' where K' is the matrix that is obtained from K by removing the row and column of entry a. So the determinant of K' is also not equal to 0, 1 or -1. This contradicts with the assumption that K is the smallest square submatrix with determinant not equal to 0, 1 or -1.

According to the fact that each row of K has at least two non-zero entries, we conclude that rows corresponding to constraints like $y_{i,j,t} \leq 1$ or $y_{i,j,t} \geq 0$ are not selected as rows of K. Therefore, without loss of generality, we can delete these rows from A, and assume that K is a submatrix of the remaining matrix. Since each column of K also has at least two non-zero entries, we can say that the columns corresponding to variables $y_{i,j,t}$ are not selected as columns of K, because each of these columns in the remaining matrix contains exactly one non-zero entry. Similarly we can remove these columns, and assume that K is a submatrix of the remaining matrix.

Now, consider a row corresponding to a constraint of the form $\sum y_{i,j,t} = Y_{i,j}$. Since the columns of variables of the form $y_{i,j,t}$ are removed later, the row corresponding to this constraint has only one non-zero entry. Again we can remove these rows from our matrix. In the remaining matrix, the columns of variables of form $Y_{i,j}$ has only one non-zero entry, therefore we remove these columns too. The remaining matrix has only rows of constraints of form $Y_{i,j} \leq x_i + x'_j$ and columns of variables of form x_i or x'_j. Note that each row of this matrix has exactly two non-zero variable with the same sign. Partition the columns into two sets $B = \{x_1, x_2, \ldots, x_n\}$ and $C = \{x'_1, x'_2, \ldots, x'_n\}$. One of those two non-zero entries belongs to a column in set B, and the other one belongs to a column in set C. According to [25], the determinant of any square submatrix of this matrix, including K is equal to 0, 1 or -1 which is again a contradiction. Therefore we conclude A is totally unimodular, and thus, the integrality gap of the corresponding integer linear program is equal to 1. □

Using ILP 5 and the result of Lemma 4, it follows that the revenue maximization problem for singleton betting is polynomial-time solvable. We conclude this section by the following theorem.

Theorem 3 *The revenue maximization problem for the auctioneer in singleton betting can be solved in polynomial time.*

Note that linear program 6 is a small polynomial-size linear program that can be solved very efficiently in practice as well. This is different from the exponential-size linear programming formulation of Chen et al. [1] for divisible variant of the subset betting.

Now consider the case that traders can order more than one share of security. The only change we should make is that the constraints of the form $y_{i,j,t} \in \{0, 1\}$ should be replaced by $y_{i,j,t} \in \{0, 1, \ldots, C\}$ in ILP 5 where C is the number of copies of a bet that the corresponding trader orders. It is clear that the integrality gap of LP 6 remains 1 in this case.

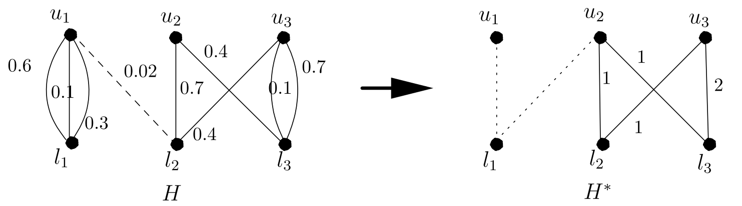

Fig. 4 Best solution with extra information

4.4 The Revenue Maximization Problem with Extra Information

In this section, we study the revenue maximization problem for singleton betting when we are given a set of pieces of extra information about the outcome of the betting market. Each piece of the extra information is a *forbidden pair* (x, y) which means that candidate x never ends up in position y. Many types of extra information could be modeled by a set of forbidden pairs. For example if we know that candidate x ends up in one of positions $y_1, \ldots, y_j$, we can model this scenario with a set of forbidden pairs (x, z), for every $z \neq y_1, \ldots, y_j$. The auctioneer may gain this type of information from various sources, or can predict such forbidden pairs with such a high confidence that he/she does not bear any risk by assuming these forbidden pairs. Let F be the set of these pairs. Given a set of forbidden pairs, a possible outcome can be illustrated by a perfect matching among candidates and positions in which no forbidden pair appears.

In that case, the auctioneer can use this information in his/her favor and in order to solve the revenue maximization problem, he/she should take into account such extra information. In this section, we show that the LP-based revenue maximization algorithm from the previous section can be extended to solve the revenue maximization problem with extra information.

Before stating the algorithm, we discuss an example. Consider the singleton betting market in Fig. 1. Suppose we know that candidates 1 and 2 do not stand at position 1 in any possible outcome (see Fig. 4). In other words, the edges $(1, 1)$ and $(2, 1)$ are forbidden pairs. Using this extra information, we can say that the candidate 3 necessarily stands at position 1. Therefore, the auctioneer gains the maximum revenue by accepting all bets except bet $(0.02, 1, 2)$. The auctioneer gets \$3.3 before the outcome, and will pay at most \$1 to the traders after the outcome.

First, we show how to calculate the minimum revenue over all possible outcomes with respect to a given accepted graph H. Then we propose a linear programming method to find, an accepted graph which maximizes this minimum revenue over the possible outcomes. Note that a possible outcome can be shown by a perfect matching M among candidates and positions that does not use the forbidden pairs. The sum of weights of edges that are in both M and H^* (see definition of G^* in Sect. 4.1) is the value that we should pay to the traders in this outcome. Therefore, in order to find the minimum revenue over all possible outcomes, we should find the maximum weighted perfect matching in H^* without using forbidden edges.

We can solve this by finding the integer solution of the following LP:

$$
\begin{aligned}
&\max\left(\sum w_{i,j}^{H^*} x_{i,j}\right)\\
&\sum_{j=1}^{n} x_{i,j} = 1 \quad \forall i \in U\\
&\sum_{i=1}^{n} x_{i,j} = 1 \quad \forall j \in L\\
&x_{i,j} = 0 \quad \forall (i,j) \in F\\
&x_{i,j} \geq 0 \quad \forall i \in U, j \in L
\end{aligned}
\tag{7}
$$

We consider the dual of the above program:

$$
\begin{aligned}
&\min\left(\sum \alpha_i + \sum \beta_j\right)\\
&\alpha_i + \beta_j \geq w_{i,j}^{H^*} \quad \forall (i,j) \notin F\\
&\alpha_i + \beta_j + \delta_{i,j} \geq w_{i,j}^{H^*} \quad \forall (i,j) \in F
\end{aligned}
\tag{8}
$$

In the above LP, α_i is the variable corresponding to the constraint of candidate $i \in U$, β_j is the variable corresponding to the constraint of position $j \in L$, and $\delta_{i,j}$ is the variable corresponding to the constraint of the forbidden edge (i, j). Note that all variables including $\delta_{i,j}$ can get arbitrarily large positive or negative values. Variables $\delta_{i,j}$ does not contribute in the cost function, so by setting $\delta_{i,j} = +\infty$, we can eliminate the constraints of the form $\alpha_i + \beta_j + \delta_{i,j} \geq w_{i,j}^{H^*}$. Therefore, we can rewrite the dual program as follows:

$$
\begin{aligned}
&\min\left(\sum \alpha_i + \sum \beta_j\right)\\
&\alpha_i + \beta_j \geq w_{i,j}^{H^*} \quad \forall (i,j) \notin F
\end{aligned}
\tag{9}
$$

This program finds the values α_i and β_j such that for each non-forbidden edge (i, j), we have $\alpha_i + \beta_j \geq w_{i,j}^{H^*}$. Since LP 7 is the similar to the LP of the maximum weighted matching problem, the integrality gap of LP 7 is 1 [24]. Therefore, the best fractional solution of the dual program 9 is equal to the maximum integer solution of LP 7.

Now we propose an algorithm to find an accepted graph maximizing this minimum revenue over the possible outcomes. With the same argument as Lemma 3, we can use values of α_i and β_j of dual program 9 to determine the value $w_{i,j}^{H^*}$ by setting it to $\min(w_{i,j}^{G^*}, \alpha_i + \beta_j)$.

We can write an integer linear program whose optimal solution determines our optimal solution for the singleton betting problem with extra information. The ILP is very similar to ILP 5. Again we want to find the values of α_i and β_j and we also want to choose edges which should be added to the optimal subgraph H.

The following program is the ILP for computing the minimum weighted vertex cover in this setting:

$$
\begin{aligned}
&\max\left(\sum w^G_{i,j,t} y_{i,j,t} - \sum x_i - \sum x'_j\right)\\
&\sum_{t=1}^{w^{G^*}_{i,j}} y_{i,j,t} = Y_{i,j} \quad \forall i \in U, j \in L\\
&Y_{i,j} \le x_i + x'_j \quad \forall (i,j) \notin F\\
&y_{i,j,t} \in \{0,1\} \quad \forall i \in U, j \in L, 1 \le t \le k_{ij}
\end{aligned}
\tag{10}
$$

Theorem 4 *The revenue maximization problem for the auctioneer in singleton betting with extra information can be solved in polynomial time.*

Proof Similar to the proof of Lemma 4, we can show that the constraint matrix of ILP 10 is totally unimodular. Therefore, if we relax the integer constraints $y_{i,j,t} \in \{0,1\}$ to linear fractional constraints $0 \le y_{i,j,t} \le 1$, we can solve this linear programming relaxation and round it to optimal integer solution of ILP 10. The optimal solution of ILP 10, corresponds to the maximum revenue of singleton betting problem with extra information. □

5 The Probabilistic Setting

In this section, we study the betting problem in a probabilistic setting. We first define the problem formally. Assume that the auctioneer has a probability distribution q over the possible outcome permutations, i.e., $\sum_{\sigma \text{ is a permutation}} q(\sigma) = 1$. Given a probability distribution q, a desired revenue x, and a desired probability $0 \le p \le 1$, we consider the following two problems:

Definition 2 In the max-expected subset betting problem, given a probability distribution q and a set I of subset bets, our goal is to find a subset S of bets I such that accepting bets in S maximizes the expected revenue of the auctioneer.

Definition 3 In the max-probability singleton betting problem, given a probability distribution q over the possible outcomes, a desired revenue x , a desired probability $0 \le p \le 1$, and a set I of simple bets, our goal is to accept a subset S of bets in I in order to have revenue x with probability at least p, and refuse to return a subset if there does not exist such a subset.

Here, we observe that max-expected subset betting problem can be solved easily, but the max-probability singleton betting problem is $\#P$-complete.

In order to solve the max-expected subset betting problem, for any subset $S \subseteq I$, let $E(S)$ be the expected revenue when the auctioneer accepts the bets in set S. For every bet $i \in I$, let $E(i)$ be the revenue when we accept only bet i. Based on the linearity of the expectation, $E(S) = E(\bigcup_{i \in S} i) = \sum_{i \in S} E(i)$. Thus, in order to maximize $E(S)$, we should add a bet $i \in I$ into S iff $E(i) > 0$. Therefore, it suffices to compute $E(i)$ for each bet i. Let p_i be the probability that the security of bet i happens. Given the probability distribution q, we can estimate p_i by sampling the probability distribution q. Thus, we can estimate $E(i) = b_i - p_i$.

Next, we prove the hardness of the max-probability singleton betting problem.

Theorem 5 *The max-probability singleton betting problem is #P-Complete.*

Proof We reduce the problem of counting the number of perfect matchings in a bipartite graph to the max-probability singleton betting problem. In fact, we consider a simpler version of this problem which is equivalent to the original one. Suppose we are given a bipartite graph G and a number k and we are asked whether there are at least k perfect matchings in G. We construct an instance of the max-probability singleton betting problem as follows. Suppose each edge in G is between sets X and Y where $|X| = |Y| = n$. For each vertex in $x_i \in X$, consider a candidate a_i in our instance. For each edge (x_i, y_j) in G, we put a bet in our instance of the form $(2, a_i, j)$ which means that this trader is willing to pay \$2 for this bet, and the trader wins \$1 if the candidate a_i stands in position j. In this instance, we set $x = 2E - n + 1$ where E is the number of edges in G. We also define p to be $1 - \frac{k-1}{n!}$. It is obvious that in the optimum solution we should accept all the bets. By definition, we are asked if the revenue is at least $2E - n + 1$ with probability $1 - \frac{k-1}{n!}$. Equivalently, we are asked if there are at most k matchings of size n in G. Thus max-probability singleton betting problem can solve the problem of calculating the number of perfect matchings in a bipartite graph in polynomial time which is a $\#P$-Complete problem. □

Note that our proof works when there are some bids (b_i, x_i, y_i) with $b_i = 2 > 1$. It is interesting to prove the same result when all bids value are between 0 and 1 which is more realistic.

6 Conclusion

In this paper, we studied the subset and singleton betting problems for permutation markets in the risk-free and probabilistic settings. We also considered the singleton betting problem with extra information in which the auctioneer has some certain knowledge of the possible outcome of the market. We showed that maximizing revenue for the subset betting problem is not approximable, but the singleton betting problem is solvable by solving a linear programming relaxation and rounding its solution, even in the presence of certain knowledge about the outcome of the market. This indicates that the betting language could play an important role in the complexity of the corresponding revenue maximization problem, which may have applications in other areas as a prediction tool.

An interesting question is to study revenue maximization problem with extra knowledge about the set of outcomes for different betting languages like pair betting and subset betting. It is also interesting to consider other types of extra information about the possible outcomes. This extra information may include some probability distribution on certain properties of the outcome. It would also be interesting to investigate the implications of the LP-based solution discussed in this paper for applications of betting markets in predication markets.

An interesting extension to our paper is to define other betting languages in which the "maximizing revenue problem" or "existence problem" could be solved in polynomial time. A suitable language for permutation markets is a language in which a trader can bet on specific subsets. For example, one of the candidates end up in the positions below or after certain position. For instance, a security can be "horse A ends up in a position better than 3." It would be interesting to study the problem for this language.

Acknowledgement The third author thank Evdokia Nikolova for introducing permutation betting markets.

Appendix: A Constructive Proof of Lemma 4

In this proof we modify an optimal fractional solution to an optimal integer solution iteratively. In fact we give an explicit polynomial-time rounding method for rounding fractional solutions of LP 6 to optimal integer solutions. In each iteration, we find some fractional variable and modify the solution based on fractional variable value and its position in the graph without any changes in objective function of LP 6.

In LP 6, we know that all values $w^G_{i,j,t}$ are nonnegative. So, in any of its optimal solutions, we have $Y_{i,j} = \min(w^{G^*}_{i,j}, x_i + x'_j)$, $y_{i,j,t} = 1$ for $1 \le t \le \lfloor Y_{i,j} \rfloor$, $y_{i,j,t} = 0$ for $t > \lfloor Y_{i,j} \rfloor + 1$, and $y_{i,j,t} = Y_{i,j} - \lfloor Y_{i,j} \rfloor$ for $t = \lfloor Y_{i,j} \rfloor + 1$. Note that the sequence $w^G_{i,j,t}$ is sorted in decreasing order with respect to their values. Consider an optimal fractional solution. There exists at least one x_i or one x'_j which is not integer. The reason is that if all values of x_i and x'_j were integer, all the other variables would be integer too. Without loss of generality, assume that x_s is not an integer number. Define critical, empty and full edges in an optimal solution of LP as follows:

- $e = (i, j)$ is a *critical* edge if we have $Y_{i,j} = x_i + x'_j = w^{G^*}_{i,j}$,
- $e = (i, j)$ is an *empty* edge if we have $Y_{i,j} = x_i + x'_j < w^{G^*}_{i,j}$, and
- $e = (i, j)$ is a *full* edge if we have $Y_{i,j} = w^{G^*}_{i,j} < x_i + x'_j$.

Consider the set of vertices in the upper (lower) part of G which have a path to u_s through the critical edges. Name this set of vertices C_U (C_L). For any critical edge $e = (i, j)$, we know that $x_i + x'_j = w^{G^*}_{i,j}$ which is an integer number, so if one of the numbers x_i and x'_j is not integer, the other one is not an integer number either. Since x_s is not an integer number, numbers x_i and x'_j are not integer where $u_i \in C_U$ and $l_j \in C_L$. This means that x_i and x'_j are greater than zero, and if we change them by a sufficiently small ε in the optimal solution, the constraints $x_i, x'_j \geq 0$ still hold. Now, we modify the optimal solution as follows:

$$x_i = x_i + \varepsilon, \quad i \in C_U$$
$$x'_j = x'_j - \varepsilon, \quad j \in C_L$$

We study in detail the changes that occur in the optimal solution by these changes of the values of these variables. First, we define ε to be zero, and then we slowly increase it and observe the changes in the solution. In the following, we study the situation of edges in different categories. In some cases, we say that we stop increasing ε if some certain events occur. Note that we keep increasing ε until at least one of these events happen, or one of the variables x_i or x'_j becomes integer.

- *All edges between C_U and C_L*: For edge $e = (i, j)$, the value of $x_i + x'_j$ does not change in this case, so nothing changes for this edge.
- *All full edges between C_U and $(L^G - C_L)$*: A full edge $e = (i, j)$ which is adjacent to C_U remains full by increasing ε. The reason is that $Y_{i,j} = w^{G^*}_{i,j} < x_i + x'_j + \varepsilon$.
- *All full edges between $(U^G - C_U)$ and C_L*: We keep increasing ε until the equality $Y_{i,j} = w^{G^*}_{i,j} = x_i + x'_j - \varepsilon$ holds for some full edges such as $e = (i, j)$ which is adjacent to C_L. In fact we stop increasing ε whenever some full edges change to critical edges.
- *All empty edges between C_U and $(L^G - C_L)$*: Consider an empty edge $e = (i, j)$ which is adjacent to C_U. Since the value of x_i increases by ε, the value of $Y_{i,j} = x_i + x'_j$ also increases. We stop increasing ε whenever some empty edges convert to critical edges. We may stop increasing ε in another situation. While we are increasing ε, the value of $Y_{i,j}$ also increases for any empty edge $e = (i, j)$ which is adjacent to C_U. By increasing $Y_{i,j}$, the value of $y_{i,j,t}$ will increase too, where t is the first index with $y_{i,j,t} < 1$. When a variable $y_{i,j,t} < 1$ reaches value 1, we stop increasing ε. Note that increasing ε can be stopped by this kind of edge in two different situations: when an empty edge changes to a critical one, or a non-integer variable $y_{i,j,t}$ becomes 1. One can see that these increments increase the objective function as well. Assume that the rate of increasing objective function by this type of change is R_1.
- *All empty edges between $(U^G - C_U)$ and C_L*: Consider an empty edge $e = (i, j)$ which is adjacent to C_L. The value of $Y_{i,j}$ decreases by increasing ε. Since we $\sum_k y_{i,j,k} = Y_{i,j}$, by decreasing $Y_{i,j}$ the value of $y_{i,j,t}$ also decreases, where t is the maximum index with a nonzero value. We stop increasing ε whenever a variable $y_{i,j,t} > 0$ reaches the value 0. These changes decrease the objective function too. Assume that the rate of decreasing the objective function by this type of changes is R_2.
- *All edges between $(U^G - C_U)$ and $(L^G - C_L)$*: For edge $e = (i, j)$, the value of $x_i + x'_j$ does not change in this case, so nothing changes for this edge.

One can verify that when we increase ε slowly the amount of change in the objective function is exactly:

$$\varepsilon R = \varepsilon |C_U| - \varepsilon |C_L| + \varepsilon R_1 - \varepsilon R_2$$

If we have $R > 0$, we can increase the objective function by increasing ε, which contradicts our optimality assumption. If we have $R < 0$, we also reach a contradiction, because if we start with $\varepsilon = 0$, and decrease its value instead, we can prove similarly, and everything is similar to the case $\varepsilon > 0$. Therefore, R should be zero. In this case, we will increase ε from zero until one of these scenarios occurs:

1. One of the full edges adjacent to C_L changes to a critical edge,
2. One of the empty edges adjacent to C_U changes to a critical edge,
3. One variable $y_{i,j,t} < 1$ reaches the value 1,
4. One variable $y_{i,j,t} > 0$ reaches the value 0, or
5. One of the non-integer variables x_i or x'_j reaches an integer value. Note that we only modify some non-integer variables x_i and x'_j in our algorithm.

If scenarios 1 or 2 occur, the number of critical edges will increase, and we can continue the algorithm iteratively with the new sets C_U and C_L. If scenarios 3 or 4 occur, the only change is the sets C_U and C_L, and we can continue our algorithm with the new sets C_U and C_L. If scenario 5 occurs, the number of integer variables x_i and x'_j will increase, and we can continue our algorithm if any non-integer variable remains. It is clear that during the changes, we have the invariant $R = 0$ in all cases. Otherwise, we can increase the objective function which is a contradiction. During these changes, we stop the algorithm if we reach an integer solution.

We now prove that our algorithm runs in polynomial time. One can see that increasing ε in our algorithm is done step by step. A step starts by increasing ε, and ends when one of scenarios 1–5 occur. We say a step is a *gold step* if it ends with scenario 5. In other words, the number of integer variables x_i and x'_j increases at the end of a gold step. It is easy to see that the number of gold steps is at most $|U^G| + |L^G|$. So, we only need to prove that the number of steps between two consecutive gold steps is a polynomial function of the input size. First, we observe that the number of steps which end with scenario 1 or 2 is polynomial. The reason is that the number of critical edges increases in these scenarios, and this number is upper bounded by the number of edges. The number of steps which end with scenarios 3 or 4 is also polynomial, since at each step one of the y variables become integral. In addition, we know that the above increasing process stops for a value $\varepsilon < 1$ because before ε reaches 1, scenario 5 occurs, so each variable $y_{i,j,t}$ triggers the conditions of scenarios 3 or 4 at most one time. Therefore, the number of steps between two consecutive gold steps is also polynomial. We can thus conclude that our algorithm runs in polynomial time. Since we do not lose any value in the modification steps, it is clear that at the end of the algorithm, we have an integer solution with the same value of the objective function.

References

1. Chen, Y., Fortnow, L., Nikolova, E., Pennock, D.: Betting on permutations. In: Proceedings of the ACM Conference on Electronic Commerce (2007)
2. Plott, C., Sunder, S.: Efficiency of experimental security markets with insider information: An application of rational expectations models. J. Polit. Econ. **90**, 663–698 (1982)
3. Plott, C., Sunder, S.: Rational expectations and the aggregation of diverse information in laboratory security markets. Econometrica **56**, 1085–1118 (1988)

4. Berg, J.E., Forsythe, R., Nelson, F.D., Rietz, T.A.: Results from a dozen years of election futures markets research. In: Plott, C.A., Smith, V. (eds.) Handbook of Experimental Economic Results. Elsevier, North-Holland (2001)
5. Forsythe, R., Rietz, T.A., Ross, T.W.: Wishes, expectations, and actions: A survey on price formation in election stock markets. J. Econ. Behav. Organ. **39**, 83–110 (1999)
6. Pennock, D.M., Lawrence, S., Giles, C.L., Nielsen, F.A.: The real power of artificial markets. Science **291**, 987–988 (2002)
7. Dhamdhere, K., Goyal, V., Ravi, R., Singh, M.: How to pay, come what may: Approximation algorithms for demand-robust covering problems. In: Proceedings of the Annual Symposium on Foundations of Computer Science (FOCS) (2005)
8. Ghodsi, M., Mahini, H., Mirrokni, V.S., Zadimoghaddam, M.: Permutation betting markets: singleton betting with extra information. In: Proceedings of the ACM Conference on Electronic Commerce (2008)
9. Nikulin, Y.: Robustness in combinatorial optimization and scheduling theory: An annotated bibliography. Technical Report SOR-91-13, Statistics and Operation Research (2004)
10. Feige, U., Jain, K., Mahdian, M., Mirrokni, V.S: Robust combinatorial optimization with exponential number of scenarios. In: Proceedings of the Conference on Integer Programming and Combinatorial Optimization (2007)
11. Immorlica, N., Karger, D., Minkoff, M., Mirrokni, V.S.: On the costs and benefits of procrastination: Approximation algorithms for stochastic combinatorial optimization problems. In: Proceedings Annual ACM-SIAM Symposium on Discrete Algorithms (SODA) (2004)
12. Gupta, A., Pal, M., Ravi, R., Sinha, A.: Boosted sampling: Approximation algorithms for stochastic optimization. In: Proceedings of the Annual ACM Symposium on Theory of Computing (STOC) (2004)
13. Shmoys, D., Swamy, S.: Stochastic optimization is (almost) as easy as deterministic optimization. In: Proceedings of the Annual Symposium on Foundations of Computer Science (FOCS) (2004)
14. Pennock, D.: Personal communications
15. Fortnow, L., Kilian, J., Pennock, D., Wellman, M.P.: Betting boolean-style: A framework for trading in securities based on logical formulas. Decis. Support Syst. **39**(1), 87–104 (2004)
16. Hanson, R.D.: Combinatorial information market design. Inf. Syst. Front. **5**(1), 107–119 (2003)
17. Cramton, P., Shoham, Y., Steinberg, R.: Combinatorial Auctions. MIT Press, Cambridge (2005)
18. Nisan, N., Roughgarden, T., Tardos, E., Vazirani, V.: Algorithmic Game Theory. Cambridge University Press, Cambridge (2007)
19. Lehmann, B., Lehmann, D., Nisan, N.: Combinatorial auctions with decreasing marginal utilities. In: Proceedings of the ACM Conference on Electronic Commerce (EC) (2001)
20. Sandholm, T.: Algorithm for optimal winner determination in combinatorial auctions. Artif. Intell. **135**, 1–54 (2002)
21. Feige, U.: On maximizing welfare when utility functions are subadditive. In: Proceedings of the Annual ACM Symposium on Theory of Computing (STOC) (2006)
22. Schrijver, A.: Total dual integrality of matching forest constraints. Combinatorica **20**, 575–588 (2000)
23. Papadimitriou, C., Steiglitz, K.: Combinatorial Optimization: Algorithms and Complexity. Dover, New York (1998)
24. Schrijver, A.: Combinatorial Optimization—Polyhedra and Efficiency. Springer, Berlin (2003)
25. Heller, I., Tompkins, C.B.: An extension of a theorem of Dantzig's. In: Kuhn, H.W., Tucker, A.W. (eds.) Linear Inequalities and Related Systems, pp. 247–254. Princeton University Press, Princeton (1956)