Achieving Proportional Representation in Conference Programs

Ioannis Caragiannis
University of Patras &
CTI 'Diophantus', Greece
caragian@ceid.upatras.gr

Laurent Gourvès
CNRS, Université
Paris-Dauphine, France
laurent.gourves@dauphine.fr

Jérôme Monnot
CNRS, Université
Paris-Dauphine, France
jerome.monnot@dauphine.fr

Abstract

We study the optimization problem of designing the program of a conference with parallel sessions, so that the intended participants are as happy as possible from the talks they can attend. Interestingly, this can be thought of as a two-dimensional extension of a scheme proposed by Chamberlin and Courant [1983] for achieving proportional representation in multi-winner elections. We show that different variations of the problem are computationally hard by exploiting relations of the problem with well-known hard graph problems. On the positive side, we present polynomial-time algorithms that compute conference programs that have a social utility that is provably close to the optimal one (within constant factors). Our algorithms are either combinatorial or based on linear programming and randomized rounding.

1 Introduction

Motivated by research on multi-winner election schemes that has been published in AI venues recently, we consider the design of the program of a scientific conference as a social choice problem. We assume that a conference center has been booked to host the conference and the conference duration is divided into time slots; a time slot coincides with the duration of a single talk. The conference center has a number of available rooms, so that different talks can take place in parallel in each time slot. The talks will be selected from contributed papers and the objective is to accept those papers, so that corresponding talks can be scheduled within the available space and time frame in such a way that the intended audience finds the program as attractive as possible.

In order to make the term "attractive" more precise, assume that the satisfaction of an intended participant is given by a non-negative *utility* she has for attending each of the contributed talks. A naive idea for optimizing the total utility (or *social utility*) of the intended audience is to select the very best papers (i.e., those that maximize the sum of utilities of the audience) among the submitted papers and fit them in the available time slots and rooms arbitrarily. This can result in a program that does not satisfy the audience as much as it could. For example, by scheduling talks on the top papers (papers that everybody likes) of the conference in parallel is a very bad choice. Indeed, each participant can attend only one of these talks and her value from the particular time slot is limited to her utility on only one of the talks in this time slot. This indicates that one should try to spread the papers with high social utility across different time slots so that each participant finds the talks she likes the most in different time slots and she is able to attend them. On the other hand, the number of time slots is limited and this makes our *conference program design* problem a computational challenge.

The problem can be thought of as an extension of a multi-winner election scheme proposed by Chamberlin and Courant [1983]. In this scheme, there is a set of alternatives (e.g., candidates for serving on a committee) and a set of voters with utilities (expressed as cardinal or ordinal preferences) over the alternatives. The goal is to elect a predefined number of alternatives and assign each voter to one of the elected alternatives (that will act as the representative of the voter) so that the total utility of the votes from their representatives is maximized. In this way, a kind of *proportional representation* of the voters is achieved in the elected set of alternatives; this has inspired the title of the current paper. To see the analogy with the conference program design problem, it suffices to consider a conference with a single time slot (and many parallel sessions). Here, the papers are the alternatives, the intended participant are the voters and the representative of each voter is the talk that each intended participant will attend.

Many papers have focused on multi-winner voting rules and proportional representation schemes recently. The ones that have received the most of attention are the schemes of Chamberlin and Courant [1983] and Monroe [1995]. As these schemes are computationally hard to resolve — a statement that has been formalized by Procaccia *et al.* [2008] — several papers have focused either on efficient (i.e., polynomial-time) *approximation algorithms* of the original rules so that social utility is provably close to the optimal one or on their exact resolution when utilities have a particular structure. For the Chamberlin-Courant scheme, Lu and Boutilier [2011] observe that the utility of the agents on their representatives in a set of alternatives forms a monotone submodular set function and resorted to a greedy algorithm (for maximizing such functions subject to a cardinality constraint) which, according to the analysis of Nemhauser *et al.* [1978], computes $(1-1/e)$-approximate solutions in polynomial time.

For more structured utilities (e.g., Borda-like utilities), Skowron *et al.* [2015a] present a polynomial-time approximation scheme. An interesting case is that of binary utilities (later, we call such utilities *uniformly dichotomous*). Then, the problem is equivalent to MAX COVER, which is well-known to be inapproximable within a factor better than $1-1/e$ in polynomial time [Feige, 1998]. Hence, the positive result of Lu and Boutilier [2011] for general utilities is best possible. Skowron and Faliszewski [2015] present improved approximations by exponential-time and FPT algorithms for this case. For Monroe's scheme — which is more distantly related to our setting — and Borda-like utilities, approximation algorithms with constant approximation ratios are presented in [Skowron *et al.*, 2015a].

So, multi-winner voting schemes belong to an area within the field of *computational social choice* [Brandt *et al.*, 2016] where approximation algorithms have found many applications. Other members of this area include the also hard-to-resolve single-winner rules by Kemeny [Ailon *et al.*, 2008; Coppersmith *et al.*, 2010; Kenyon-Mathieu and Schudy, 2007] and Dodgson [Caragiannis *et al.*, 2012; 2014]. Other recent papers related to Chamberlin-Courant scheme study its parametrized complexity [Betzler *et al.*, 2013] and its complexity in restricted domains [Skowron *et al.*, 2015b] and online settings [Oren and Lucier, 2014].

Clearly, conference program design is computationally hard as a generalization of the Chamberlin-Courant scheme. We argue that it is the additional structure required by the solution (talks organized in parallel sessions and time slots) that makes the problem hard even in extensions of trivial Chamberlin-Courant elections. One such case is when all the contributed papers fit in the program and paper selection is not necessary. In Section 3, we show that instances of this kind with uniformly dichotomous utilities are APX-hard for programs with two time slots and NP-hard for program with three parallel sessions. Our proofs use reductions from the well-known graph problems MAX BISECTION and PARTITION INTO TRIANGLES.

Then, we present approximation algorithms, i.e., polynomial-time algorithms that design conference programs in which the social utility is provably close to the social utility of the optimal conference program. This turns out to be a very challenging task. Unlike in the case of Chamberlin-Courant, our problem does not correspond to submodular function maximization any more (at least, not in any way that is apparent to us) and greedy solutions either seem too complicated to analyse or have obvious counter-examples with poor social utility. So, we have resorted to non-greedy techniques for designing approximation algorithms.

In Section 4, we present a combinatorial deterministic algorithm, which uses a *graph matching* computation and solves the problem exactly (i.e., computes a program of maximum social utility) for conferences with two parallel sessions; this algorithm is used to approximate the problem in the case of more sessions. The *approximation ratio* (i.e., the worst-case social utility guarantee of the algorithm compared to the solution of maximum social utility) grows linearly with the number of sessions. So, we present two more approximation algorithms that do not have this drawback and achieve *constant* approximation ratios. Our second algorithm runs in polynomial time when the number of parallel sessions is constant (Section 5) while the third algorithm does not have any such restriction but achieves a slightly worse approximation ratio (Section 6). Both our second and third algorithm are based on the paradigm of *linear programming* and *randomized rounding* that has been very successful in the design of approximation algorithms (the book by Williamson and Shmoys [2011] provides a nice coverage of the field).

We continue with preliminary definitions; a discussion on open problems and extensions can be found in Section 7.

2 Definitions

We abbreviate our problem as CPD (standing for conference program design). Instances of CPD consist of a set L of n *agents*, a set X of m *items*, two positive integers k and q such that $m \geq kq$, and a *utility function* $u_\ell : X \rightarrow \mathbb{R}_{\geq 0}$ for each agent $\ell \in L$. A *solution* $\mathcal{S}$ is a collection of k disjoint subsets $S_1, ..., S_k$ of X, with $|S_i| = q$ for $i \in [k] = \{1, 2, ..., k\}$. Agents enjoy their most preferred item in each set S_i and have *additive* utilities; the utility of agent ℓ for $\mathcal{S}$ is defined as

$$u_\ell(\mathcal{S}) = \sum_{i=1}^{k} \max_{x \in S_i} u_\ell(x).$$

The social utility is defined as $\mathsf{U}(\mathcal{S}) = \sum_{\ell \in L} u_\ell(\mathcal{S})$. On input an instance $(L, X, (u_\ell)_{\ell \in L}, k, q)$, the objective of CPD is to compute a solution $\mathcal{S}$ that maximizes $\mathsf{U}(\mathcal{S})$. In brief:

Conference program design (CPD)	
Input:	A set L of n agents, a set X of m items, positive integers k and q such that $m \geq kq$, and a utility function $u_\ell : X \rightarrow \mathbb{R}_{\geq 0}$ for each agent $\ell \in L$
Output:	A collection $\mathcal{S}$ of k disjoint subsets $S_1, ..., S_k$ of X, each of size q
Goal:	To maximize the quantity $\mathsf{U}(\mathcal{S}) = \sum_{\ell \in L} \sum_{i=1}^{k} \max_{x \in S_i} u_\ell(x)$

The analogy of CPD to the conference scenario is as follows. Each item corresponds to a contributed paper. The parameters q and k indicate the number of parallel sessions and the number of time slots, respectively. A solution consisting of k disjoint q-sized sets of items should be thought of as sets of q talks that can be given in parallel in k available time slots. Agents correspond to intended participants. In each time slot, a participant can attend one talk and obtain a utility from that talk only. Naturally, in each time slot, an agent will get the highest utility among the available talks.

As an example, suppose we have 7 items i_1, i_2, ..., i_7 and 3 agents a_1, a_2, and a_3 with the utilities in the table below:

	i_1	i_2	i_3	i_4	i_5	i_6	i_7
a_1	4	3	5	1	2	0	4
a_2	1	4	3	9	6	2	0
a_3	6	1	2	0	0	4	5

We have $k = 3$ time slots and $q = 2$ parallel sessions. Consider the solution $\mathcal{S} = (\{i_3, i_6\}, \{i_4, i_7\}, \{i_1, i_5\})$ indicating that the talks corresponding to items i_3 and i_6 are scheduled in the two parallel sessions of the first time slot, the talks corresponding to i_4 and i_7 in the second time slot, and the talks corresponding to i_1 and i_5 in the third time slot. Using our definitions, the utility of agent a_1 from the solution $\mathcal{S}$ is the sum of her utility on the three time slots, i.e., $u_1(\mathcal{S}) = u_1(\{i_3, i_6\}) + u_1(\{i_4, i_7\}) + u_1(\{i_1, i_5\}) = \max\{5, 0\} + \max\{1, 4\} + \max\{4, 2\} = 13$ and, similarly, we obtain $u_2(\mathcal{S}) = 18$ and $u_3(\mathcal{S}) = 15$ for a social utility of $\mathsf{U}(\mathcal{S}) = u_1(\mathcal{S}) + u_2(\mathcal{S}) + u_3(\mathcal{S}) = 46$.

Note that each item can appear in at most one set and there can be items that do not belong to any set of the solution (e.g., item i_2 in the example). Also, note that the problem does not constrain in any way the assignment of items (talks) to rooms or the order of the time slots. For example, the social utility of solution $\mathcal{S}$ does not depend at all on whether the set of items $\{i_3, i_6\}$ is scheduled before the set $\{i_4, i_7\}$ or not.

In the following, we sometimes distinguish between two different types of CPD instances. In those with $m = kq$, the available items fit in the program and the question is just to partition them into k equally sized sets in order to form the final solution. Our hardness results apply on such instances. Another assumption made there is that of uniformly dichotomous utilities that take values in $\{0, 1\}$. Our algorithmic results in Sections 4, 5, and 6 apply to any CPD instance (without restrictions on k and q or on the agent utilities).

3 Hardness Results

The hardness of the multi-winner rule of Chamberlin and Courant [1983] (i.e., the restriction of CPD to instances with $k = 1$) is due to the difficulty in selecting the q items that maximize the social utility among many more available items. In contrast, the two hardness results presented below apply specifically to conference programs that have enough room to accommodate all items and the question that turns out to be hard is how to schedule the items in different time slots. The first hardness result (Theorem 1) applies to conference programs with only two time slots; the second one (Theorem 2) applies to conference programs with three parallel sessions. Both statements apply to uniformly dichotomous utilities. Reductions from different problems (MAX BISECTION and PARTITION INTO TRIANGLES) are used in the proofs.

Theorem 1. *The restriction of* CPD *to instances with $k = 2$, $m = 2q$, and agents with uniformly dichotomous utilities is NP-hard to approximate within a factor of $50/51 \approx 0.9804$.*

Proof. We will use a polynomial-time approximation-preserving reduction from MAX BISECTION; this is the optimization version of the well-known NP-hard decision problem [GT16] from the book of Garey and Johnson [1979]).

MAX BISECTION	
Input:	A simple graph $G = (V, E)$ with $2n$ nodes
Output:	A partition of V into two sets V_1, V_2 with $\lvert V_1\rvert = \lvert V_2\rvert = n$
Goal:	To maximize the number of edges with one endpoint in V_1 and the other one in V_2

Starting from an instance of MAX BISECTION consisting of a graph $G = (V, E)$ with an even number of nodes, we construct an instance $\mathcal{I}$ of CPD that has an item for each node of G (i.e., $X = V$), an agent $a_{vv'}$ for every edge $(v, v') \in E$ with utility 1 for items v and v' and 0 for every other item, and by setting the two parameters to $k = 2$ and $q = |V|/2$. Clearly, the utilities of instance $\mathcal{I}$ are uniformly dichotomous; the construction is polynomial-time.

Observe that there is a one-to-one correspondence between bisections of G and solutions to $\mathcal{I}$. A bisection $\langle V_1, V_2\rangle$ of G has the corresponding solution $\mathcal{S} = (V_1, V_2)$ of $\mathcal{I}$. The utility of an agent $a_{vv'}$ that corresponds to an edge (v, v') with its endpoints in different sides (respectively, in the same side) of the bisection $\langle V_1, V_2\rangle$ is 2 (respectively, 1). Hence,

$$|\langle V_1, V_2\rangle| = \mathsf{U}(\mathcal{S}) - |E|. \tag{1}$$

Due to the one-to-one correspondence between solutions of the two problems, equation (1) holds for the optimal solutions of the two problems, namely the size of the maximum bisection $\langle V_1^*, V_2^*\rangle$ and the optimal social utility $\mathsf{U}(\mathcal{S}^*)$ of $\mathcal{I}$. Thus, any polynomial-time algorithm that returns a ρ'-approximate solution $\mathcal{S}$ for CPD (i.e., $\mathsf{U}(\mathcal{S}) = \rho'\mathsf{U}(\mathcal{S}^*)$) implies a ρ-approximate solution for MAX BISECTION such that

$$\rho = \frac{|\langle V_1, V_2\rangle|}{|\langle V_1^*, V_2^*\rangle|} = \frac{\mathsf{U}(\mathcal{S}) - |E|}{|\langle V_1^*, V_2^*\rangle|} = \frac{\rho'\mathsf{U}(\mathcal{S}^*) - |E|}{|\langle V_1^*, V_2^*\rangle|}$$
$$= \frac{\rho'|\langle V_1^*, V_2^*\rangle| - (1-\rho')|E|}{|\langle V_1^*, V_2^*\rangle|} \geq 3\rho' - 2,$$

where the last inequality follows by the well-known fact (e.g., see Lee *et al.* [2013]) that the optimal bisection includes at least half of the edges in a graph. So, any result stating that approximating MAX BISECTION within a factor of ρ or better is NP-hard implies that approximating CPD within a factor of $\frac{2+\rho}{3}$ or better is NP-hard. The theorem follows by applying the classical inapproximability bound of $16/17$ for (MAX CUT and, consequently, for) MAX BISECTION due to Håstad [2001]. □

A strengthened analysis (omitted due to lack of space) yields an inapproximability bound of approximately 0.944, assuming the *Unique Games Conjecture* and using the hardness results of Khot *et al.* [2007].

Theorem 2. *The restriction of* CPD *to instances with $q = 3$, $m = 3k$, and agents with dichotomous utilities is NP-hard.*

Proof. We will prove the theorem using a polynomial-time reduction from PARTITION INTO TRIANGLES (problem [GT11] in [Garey and Johnson, 1979]), defined as follows:

PARTITION INTO TRIANGLES	
Input:	A graph $G = (V, E)$, with $\lvert V\rvert = 3p$ for some integer p
Question:	Can V be partitioned into p disjoint sets $V_1, V_2, ..., V_p$, each containing exactly three nodes, such that for every $i \in [p]$, all the three edges (u_i, v_i), (u_i, w_i) and (w_i, v_i) belong to E?

Starting from an instance $\mathcal{I}$ of PARTITION INTO TRIANGLES (i.e., a graph $G = (V, E)$ with $3p$ nodes), we build an instance $\mathcal{I}' = (X, L, (u_\ell)_{\ell \in L}, k, q)$ of CPD as follows. Instance $\mathcal{I}'$ has an item for each node in G, i.e., $X = V$. For each pair of nodes $v, v' \in V$ such that $(v, v') \notin E$, $\mathcal{I}'$ has an agent $a_{vv'}$ who has utility 1 for items v and v' and utility 0 for every other item $v'' \in X \setminus \{v, v'\}$. The two parameters are set to $k = p$ and $q = 3$. Clearly, the number of items is $3k$ and the utilities are uniformly dichotomous.

The reduction is certainly polynomial-time. We claim that there is a solution of $\mathcal{I}'$ with social utility $2\left(\binom{3p}{2} - |E|\right)$ if and only if $\mathcal{I}$ is a "yes" instance.

Indeed, given a solution $\mathcal{S}$ of $\mathcal{I}'$ with social utility $2 \cdot \left(\binom{3p}{2} - |E|\right)$, let $V_1, ..., V_p$ be a solution of $\mathcal{I}$ such that $V_i = S_i$, $i = 1, ..., p$. The utility of an agent $a_{vv'}$ will be either 1 or 2, depending on whether the items v and v' belong to the same set of $\mathcal{S}$ or not. Since $\mathcal{I}'$ contains $\binom{3p}{2} - |E|$ agents, they all must have utility 2 in $\mathcal{S}$. This implies that, for $i = 1, ..., p$, the three edges $\{(u_i, v_i), (u_i, w_i), (w_i, v_i)\}$ belong to E since, otherwise, we would have an agent with utility 1.

Now, starting with a partition of G into triangles $V_1, ..., V_p$, let $\mathcal{S}$ be a solution of $\mathcal{I}'$ with $S_i = V_i$ for $i = 1, ..., k$. We have $u_\ell(\mathcal{S}) = 2$ for every agent ℓ because the two items that she values by 1 are in different sets. Since there are $\binom{3p}{2} - |E|$ agents, we obtain $\mathsf{U}(\mathcal{S}) = 2 \cdot \left(\binom{3p}{2} - |E|\right)$, as desired. □

4 A Combinatorial Approximation Algorithm

Our first approximation algorithm for CPD is based on an exact polynomial-time algorithm for the case $q = 2$ which in turn exploits the relation of CPD to graph matchings in this case. The main idea is to build an edge-weighted graph, in which each node corresponds to an item and an edge between two items has the total utility of all agents for the pair of items. Then, CPD is equivalent to selecting a matching with exactly k edges that has maximum total weight, since such a matching corresponds to a collection of k disjoint pairs of items and maximum total weight corresponds to maximum social utility.

More formally, on input an instance $\mathcal{I} = (X, L, (u_\ell)_{\ell \in L}, k, q)$ with $q = 2$ and $m \geq 2k$, our algorithm constructs a graph G with m nodes corresponding to the items in X (*item nodes*) and $m - 2k$ additional *dummy nodes*. There is an edge (x, x') in G between every pair of item nodes x and x' and an edge (x, y) between every item node x and every dummy node y. The weight of an edge (x, x') between item nodes is set equal to $\sum_{\ell \in L} \max\{u_\ell(x), u_\ell(x')\}$, the total utility of all agents from the pair of items $\{x, x'\}$. The weight of an edge (x, y) between an item node and a dummy node is 0. The algorithm computes a perfect matching of maximum weight using a (polynomial-time) implementation of the algorithm of Edmonds [1965a; 1965b] (e.g., see Gabow [1990]). Since an edge of the graph has at most one dummy node as an endpoint, any perfect matching of G will have exactly k edges between item nodes and $m - 2k$ edges involving one dummy node each. The item pairs corresponding to the k edges between item nodes form the final CPD solution for $\mathcal{I}$. The next statement should now be clear.

Theorem 3. *The matching-based algorithm above solves exactly* CPD *instances with $q = 2$ in polynomial-time.*

Now, given an instance $\mathcal{I}$ of CPD with $q > 2$, we first use the algorithm above to compute the optimal solution $\mathcal{T}$ to the restriction $\mathcal{I}_2$ of the problem with $q = 2$. The solution to instance $\mathcal{I}$ is obtained by putting additional items into each set of $\mathcal{T}$ so that each set has size q and no item appears in two different sets. We will show that this yields a $2/q$-approximation, proving the next theorem.

Theorem 4. *The matching-based algorithm described above is a polynomial-time $2/q$-approximation algorithm for instances of* CPD *with $q \geq 2$.*

Proof. Since the algorithm computes an optimal solution $\mathcal{T}$ for the restriction $\mathcal{I}_2$ of instance $\mathcal{I}$ for $q = 2$, it suffices to show that $\mathcal{T}$ has social utility at least $2/q$ times the optimal social utility for $\mathcal{I}$. Indeed, let $\mathcal{S}$ be an optimal CPD solution for instance $\mathcal{I}$ and consider any set S_i (for $i \in [k]$) in $\mathcal{S}$. We claim that there exist a pair T_i of items (to be thought of as part of a solution for $\mathcal{I}_2$) in S_i such that

$$\sum_{\ell \in L} \max_{x \in T_i} u_\ell(x) \quad \geq \quad \frac{2}{q} \sum_{\ell \in L} \max_{x \in S_i} u_\ell(x). \tag{2}$$

The desired result will then follow by summing inequality (2) over all $i \in [k]$ and using the fact that the additional items put into the sets of T do not decrease the social utility.

To see why (2) is true, let $\text{top}_\ell(S_i)$ denote the item of S_i for which agent ℓ has highest utility, breaking possible ties arbitrarily. Then, $\sum_{\ell \in L} \max_{x \in S_i} u_\ell(x) = \sum_{x \in S_i} \sum_{\ell \in L: \text{top}_\ell(S_i) = x} u_\ell(x)$, i.e., we have expressed the total utility of the agents as a sum of utilities over the q items in S_i. Clearly, there are two terms in this sum that contribute at least a fraction of $2/q$ to it, and it suffices to use the corresponding two items to form set T_i. Hence, $\sum_{x \in T_i} \sum_{\ell \in L: \text{top}_\ell(S_i) = x} u_\ell(x) \geq \frac{2}{q} \sum_{\ell \in L} \max_{x \in S_i} u_\ell(x)$, which is equivalent to (2). □

5 An LP-based Approximation Algorithm

Our next algorithm is based on linear programming and randomized rounding, following a very successful approach in the field of approximation algorithms for computationally hard combinatorial optimization problems [Williamson and Shmoys, 2011]. It uses a natural formulation of CPD as an *integer linear program*, solves efficiently its *linear programming relaxation* to obtain a fractional solution, and then rounds the fractional solution into an integral one which can be proved to be close to the optimal CPD solution.

Denote by $\mathcal{Q}$ the collection of q-sized subsets of X. For an item $i \in X$, let $\mathcal{Q}_i$ be the sets of $\mathcal{Q}$ that contain item i. CPD is equivalent to the integer linear program:

$$\begin{aligned} \text{maximize} \quad & \sum_{\ell \in L} \sum_{S \in \mathcal{Q}} \mathbf{x}(S) \cdot u_\ell(S) \qquad (3) \\ \text{subject to:} \quad & \sum_{S \in \mathcal{Q}_i} \mathbf{x}(S) \leq 1, \forall i \in X \\ & \sum_{S \in \mathcal{Q}} \mathbf{x}(S) = k \\ & \mathbf{x}(S) \in \{0, 1\}, \forall S \in \mathcal{Q} \end{aligned}$$

ILP (3) has a binary variable $\mathbf{x}(S)$ for each set of $\mathcal{Q}$ that indicates whether set S is part of the CPD solution ($\mathbf{x}(S) = 1$) or not ($\mathbf{x}(S) = 0$). The first constraint of (3) requires that each item appears in at most one set in the CPD solution which, by the second constraint, consists of exactly k sets. The objective is to maximize to social utility, expressed using the binary variables.

Our algorithm solves the relaxed version of (3) where the integrality constraint $\mathbf{x}(S) \in \{0, 1\}$ is replaced by the inequality $\mathbf{x}(S) \geq 0$. So, the variables in the relaxed version can take any (fractional) value between 0 and 1 (values higher than 1 would violate the first set of constraints). Observe that, if q is a constant, the number of variables is polynomial (in terms of the number of items and, consequently, in terms of the number of bits in the representation of the input); hence, the linear program can be solved in polynomial time.

Then, using the fractional solution $\mathbf{x}$ of the LP relaxation of (3), the algorithm performs the following randomized rounding procedure. For $j = 1, 2, ..., k$, it casts a die having a face for each set S of $\mathcal{Q}$ with associated probability $\mathbf{x}(S)/k$. The set corresponding to the die casting outcome is temporarily selected to be the j-th set in the integral solution. So far, there is no guarantee that the same item does not appear more than once (i.e., in different sets selected) and an elimination step is executed to produce a feasible integral solution: for each item i that appears in more than one sets that were selected, one of these sets is selected uniformly at random (this will be the only set in which item i will appear in the final integral solution) and item i is removed from the other sets. The final CPD solution is obtained by putting additional items into each set so that each set has exactly q items and no item appears in two different sets. The analysis of the algorithm follows.

Theorem 5. *Given an instance of* CPD *with constant $q \geq 2$, the LP-based algorithm above runs in polynomial time and computes a solution with expected social utility that is at least $1 - 1/e \approx 0.6321$ times the optimal social utility.*

Proof. Denote by $\mathbf{x}$ the fractional solution of the LP and by $\hat{\mathbf{x}}$ the final integral solution. By overloading notation, we define two (artificial) quantities that will be useful in the proof. In particular, the utility of agent $\ell \in L$ from a fractional solution $\mathbf{x}$ is defined as $u_\ell(\mathbf{x}) = \sum_{S\in\mathcal{Q}} \mathbf{x}(S) \cdot u_\ell(S) = \sum_{S\in\mathcal{Q}} \mathbf{x}(S) \cdot \max_{i\in S} u_\ell(i)$ and the social utility of a fractional solution $\mathbf{x}$ is defined as $\mathsf{U}(\mathbf{x}) = \sum_{\ell\in L} u_\ell(\mathbf{x})$.

Our main goal is to prove that

$$\mathbb{E}[u_\ell(\hat{\mathbf{x}})] \quad \geq \quad (1 - 1/e) \cdot u_\ell(\mathbf{x}) \tag{4}$$

for every agent $\ell \in L$. The claimed approximation guarantee will then follow easily. Indeed, summing inequality (4) over all agents and using linearity of expectation, we will get

$$\begin{aligned}\mathbb{E}[\mathsf{U}(\hat{\mathbf{x}})] &= \sum_{\ell\in L} \mathbb{E}[u_\ell(\hat{\mathbf{x}})] \geq (1 - 1/e) \sum_{\ell\in L} u_\ell(\mathbf{x}) \\ &\geq (1 - 1/e) \sum_{\ell\in L} u_\ell(\mathbf{x}^*) = (1 - 1/e)\mathsf{U}(\mathbf{x}^*),\end{aligned}$$

where $\mathbf{x}^*$ is the optimal solution for the given instance of CPD. Here, we have used the property that the optimal objective value of the LP relaxation of (3) is equal to $\sum_{\ell\in L} u_\ell(\mathbf{x})$ and is not lower than the objective value of (3) at the optimal integral solution $\mathbf{x}^*$.

In the following, we prove inequality (4) for an agent $\ell \in L$. For each set S of $\mathcal{Q}$, we define the critical item of S for agent ℓ to be the one of maximum utility for agent ℓ. If more than one items in a set have maximum utility for agent ℓ, we arbitrarily select one to be her critical item of the set. Denote by $\mathcal{Q}_{i,\ell}$ the subset of $\mathcal{Q}$ consisting of sets that contain item i as a critical item for agent ℓ. Now, in order to bound from below the utility of agent ℓ in the final integral solution, we will only consider the sets selected during the rounding step for which the critical item survived after the elimination step. This is expressed by the inequality

$$\mathbb{E}[u_\ell(\hat{\mathbf{x}})] \quad \geq \quad \sum_{i\in X} \sum_{S\in\mathcal{Q}_{i,\ell}} u_\ell(i) \cdot \Pr[\mathcal{E}(i, S)] \tag{5}$$

where $\mathcal{E}(i, S)$ denotes the event that set S (belonging to $\mathcal{Q}_{i,\ell}$) was selected in some of the k rounding steps and item i was not removed from it during the elimination process. Notice that sets, in which the critical item for agent ℓ did not survive after the elimination step, may still contribute to the utility of ℓ; we will simply ignore these sets in our analysis.

To conclude the proof, we observe that the event $\mathcal{E}(i, S)$ is true when

- set S is selected in the j-th rounding step (with probability $\mathbf{x}(S)/k$) for some $j \in [k]$,
- t sets (for some $t \in \{0, 1, ..., k - 1\}$) containing item i are selected in rounding steps different than j (the probability that some set of $\mathcal{Q}_i$ is selected at a given step is $\frac{1}{k}\sum_{S\in\mathcal{Q}_i} \mathbf{x}(S)$), and
- among the $t + 1$ appearances of item i in sets selected during the rounding steps, the (copy of the) set S selected at the j-th rounding step is the one selected uniformly at random (as the set in which item i will survive; obviously, the corresponding probability is $\frac{1}{t+1}$).

Using $y = \sum_{S\in\mathcal{Q}_i} \mathbf{x}(S)$ (which means that $y \in (0, 1]$ by the LP constraints), we can bound $\Pr[\mathcal{E}(i, S)]$ as follows:

$$\begin{aligned}&\Pr[\mathcal{E}(i, S)] \\ &= \sum_{j=1}^{k} \frac{\mathbf{x}(S)}{k} \sum_{t=0}^{k-1} \binom{k-1}{t} \left(\frac{y}{k}\right)^t \left(1 - \frac{y}{k}\right)^{k-1-t} \frac{1}{t+1} \\ &= \frac{\mathbf{x}(S)}{y} \sum_{t=1}^{k} \binom{k}{t} \left(\frac{y}{k}\right)^t \left(1 - \frac{y}{k}\right)^{k-t} \\ &= \frac{\mathbf{x}(S)}{y} \left(1 - \left(1 - \frac{y}{k}\right)^k\right) \\ &\geq \left(1 - (1 - 1/k)^k\right) \cdot \mathbf{x}(S) \\ &\geq (1 - 1/e) \cdot \mathbf{x}(S),\end{aligned} \tag{6}$$

where the first inequality follows since $y \in (0, 1]$ and the function $\frac{1}{y}\left(1 - (1 - y/k)^k\right)$ is non-increasing in terms of y in that range (the derivative with respect to y is non-positive) and the second one follows since $(1 - 1/k)^k \leq 1/e$ for every (integer) $k > 0$.

Using (5) and (6), and recalling that $u_\ell(i) = u_\ell(S)$ for each set S which contains item i as critical for agent ℓ, we have

$$\begin{aligned}\mathbb{E}[u_\ell(\hat{\mathbf{x}})] &\geq (1-1/e)\sum_{i\in X}\sum_{S\in\mathcal{Q}_{i,\ell}}\mathbf{x}(S)\cdot u_\ell(i)\\ &= (1-1/e)\sum_{S\in\mathcal{Q}}\mathbf{x}(S)\cdot u_\ell(S)\\ &= (1-1/e)\cdot u_\ell(\mathbf{x}),\end{aligned}$$

as desired. □

6 Handling Instances with Superconstant q

We now present a different LP-based algorithm. The main advantage is in the use of the new LP formulation (7) below. Now, the variables are not associated with sets and, consequently, the linear program (7) has polynomial size and can be solved in polynomial time even when q is unbounded. This will come at the cost of an additional multiplicative factor of $1/e$ in the approximation ratio compared to the algorithm of the previous section.

$$\begin{aligned}\text{maximize} \quad & \sum_{\ell\in L}\sum_{j\in[k]}\sum_{i\in X}\mathbf{x}_{\ell,i,j}\cdot u_\ell(i) \qquad (7)\\ \text{subject to:} \quad & \sum_{i\in X}\mathbf{x}_{\ell,i,j} = 1, \forall \ell\in L, j\in[k]\\ & x_{\ell,i,j} \leq \mathbf{y}_{ij}, \forall \ell\in L, j\in[k], i\in X\\ & \sum_{j\in[k]}\mathbf{y}_{ij} \leq 1, \forall i\in X\\ & \sum_{i\in X}\mathbf{y}_{ij} = q, \forall j\in[k]\\ & \mathbf{x},\mathbf{y}\geq 0\end{aligned}$$

Note that (7) is the relaxed version of an integer LP with binary variables that represents CPD. In our interpretation of the LP, we will use the term *level*. The solution sought by the LP is organized in k levels, numbered from 1 to k. Each level hosts a set of q items. Intuitively, the variables witness the membership of items to the (sets at the) different levels. In particular, variable $\mathbf{x}_{\ell,i,j}$ indicates whether item $i\in X$ gives agent $\ell\in L$ her maximum utility among all items in level $j\in[k]$. Variable $\mathbf{y}_{ij}$ indicates whether item $i\in X$ is in level $j\in[k]$. The first set of constraints indicates that exactly one item gives agent ℓ her highest utility at level j. The second set of constraints indicates that item i should appear at level j if it gives her maximum utility to some agent. The third set of constraints guarantees that any item i appears in at most one level. The fourth set of constraints indicates that exactly q items should appear in any level.

Randomized rounding is now performed as follows. In order to come up with an intermediate set of up to q items in each of the levels $j = 1, ..., k$, it repeats the following selection q times. It casts a die with m faces, one face for each item $i\in X$ with associated probability $\frac{1}{qk}\sum_{j\in[k]}\mathbf{y}_{ij}$. At the end of the process, the intermediate sets of items at each level consist of up to q items since an item may have been selected more than once at a given level. Also, the same item may have been allocated in more than one levels. In order to come up with a final integral solution in which each item appears only once, the following elimination step is applied. For every item i, among the several levels that contain item i, one is selected uniformly at random (this will be the only level in which item i will appear in the final integral solution) and item i is removed from the other levels. Again, the final solution is obtained by putting additional items into each set so that each set has size exactly q and no item appears in two different sets.

The proof of the next statement is omitted due to lack of space; it uses more involved arguments than those in the proof of Theorem 5 and is considerably longer.

Theorem 6. *Given an instance of* CPD*, the LP-based algorithm just described runs in polynomial time and computes a solution with expected social utility that is at least* $1/e - 1/e^2 \approx 0.2325$ *times the optimal social utility.*

7 Discussion

We conclude with a discussion on possible extensions of our work. First, we remark that even though our LP-based algorithms show that constant approximation guarantees in polynomial time are possible, they are randomized. Derandomization techniques described in classical textbooks such as [Williamson and Shmoys, 2011] might be useful in order to perform the rounding step deterministically but it would be highly desirable to achieve constant approximation bounds by combinatorial (deterministic) algorithms. This is an important problem that we have left widely open together with the following two challenging questions: Is a polynomial-time approximation scheme possible for CPD instances with constant q? For general instances, is there a combinatorial deterministic $(1-1/e)$–approximation algorithm?

Even though the conference-related terminology is used throughout the paper, applications of our setting are not limited to scientific conferences and could also include big sport events such as the Olympics or cultural events such as film or music festivals. Of course, in the corresponding optimization problems there are often additional subtle constraints such as the requirement of scheduling talks by the same speaker in different time slots. We believe that they can be easily incorporated into our setting; we have decided to neglect them here in order to showcase the relation to multi-winner voting.

A more interesting extension would be to consider capacity constraints in each session, essentially requiring that the number of intended participants that can have a talk as their most favourite one in a time slot is limited. This would lead to two-dimensional extensions of the proportional representation scheme of Monroe [1995]. Also, considering different definitions of social utility would be interesting. In this direction, we have some preliminary results on an egalitarian definition of social utility that indicate that CPD is, in general, hard-to-approximate under this objective. Complementing this statement with positive results for instances of particular structure would complete the picture nicely. Finally, practical issues such as the elicitation of utilities by the intended participants deserve much investigation.

Acknowledgements

Caragiannis was partially supported by the Caratheodory research grant E.114 from the University of Patras and the COST Action IC1205 on "Computational Social Choice". Gourvès and Monnot acknowledge support from the ANR project CoCoRICo-CoDec. Part of this work was done while Caragiannis was visiting Université Paris-Dauphine as an invited professor. The authors thank Jérôme Lang for helpful discussions in all stages of this work.

References

[Ailon *et al.*, 2008] N. Ailon, M. Charikar, and A. Newman. Aggregating inconsistent information: Ranking and clustering. *Journal of the ACM*, 55(5): article 23, 2008.

[Betzler *et al.*, 2013] N. Betzler, A. Slinko, and J. Uhlmann. On the computation of fully proportional representation. *Journal of Artificial Intelligence Research*, 47:475–519, 2013.

[Brandt *et al.*, 2016] F. Brandt, V. Conitzer, U. Endress, J. Lang, and A. D. Procaccia, editors. *Hanbook of Computational Social Choice*. Cambridge University Press, 2016.

[Caragiannis *et al.*, 2012] I. Caragiannis, J. A. Covey, M. Feldman, C. M. Homan, C. Kaklamanis, N. Karanikolas, A. D. Procaccia, and J. S. Rosenschein. On the approximability of Dodgson and Young elections. *Artificial Intelligence*, 187–188:31–51, 2012.

[Caragiannis *et al.*, 2014] I. Caragiannis, C. Kaklamanis, N. Karanikolas, and A. D. Procaccia. Socially desirable approximations for Dodgson's voting rule. *ACM Transactions on Algorithms*, 10(2): article 6, 2014.

[Chamberlin and Courant, 1983] J. R. Chamberlin and P. N. Courant. Representative deliberations and representative decisions: proportional representation and the Borda rule. *American Political Science Review*, 77(3):718–733, 1983.

[Coppersmith *et al.*, 2010] D. Coppersmith, L. Fleischer, and A. Rudra. Ordering by weighted number of wins gives a good ranking for weighted tournaments. *ACM Transactions on Algorithms*, 6(3): article 55, 2010.

[Edmonds, 1965a] J. Edmonds. Maximum matching and a polyhedron with 0, 1-vertices. *Journal of Research National Bureau of Standards Section B*, 69:125–130, 1965.

[Edmonds, 1965b] J. Edmonds. Paths, trees, and flowers. *Canadian Journal of Mathematics*, 17:449–467, 1965.

[Feige, 1998] U. Feige. A threshold of $\ln n$ for approximating set cover. *Journal of the ACM*, 45(4):634–652, 1998.

[Gabow, 1990] H. N. Gabow. Data structures for weighted matching and nearest common ancestors with linking. In *Proceedings of the 1st Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 434–443, 1990.

[Garey and Johnson, 1979] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, NY, USA, 1979.

[Håstad, 2001] J. Håstad. Some optimal inapproximability results. *Journal of the ACM*, 48(4):798–859, 2001.

[Kenyon-Mathieu and Schudy, 2007] C. Kenyon-Mathieu and W. Schudy. How to rank with few errors. In *Proceedings of the 39th Annual ACM Symposium on Theory of Computing (STOC)*, pages 95–103, 2007.

[Khot *et al.*, 2007] S. Khot, G. Kindler, E. Mossel, and R. O'Donnell. Optimal inapproximability results for MAX-CUT and other 2-variable CSPs? *SIAM Journal on Computing*, 37(1):319–357, 2007.

[Lee *et al.*, 2013] C. Lee, P.-S. Loh, and B. Sudakov. Bisections of graphs. *Journal of Combinatorial Theory, Series B*, 103(5):599–629, 2013.

[Lu and Boutilier, 2011] T. Lu and C. Boutilier. Budgeted social choice: From consensus to personalized decision making. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI)*, pages 280–286, 2011.

[Monroe, 1995] B. L. Monroe. Fully proportional representation. *American Political Science Review*, 89:925–940, 1995.

[Nemhauser *et al.*, 1978] G. Nemhauser, L. Wolsey, and M. Fischer. An analysis of approximations for maximizing submodular set functions. *Mathematical Programming*, 14:265–294, 1978.

[Oren and Lucier, 2014] J. Oren and B. Lucier. Online (budgeted) social choice. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence (AAAI)*, pages 1456–1462, 2014.

[Procaccia *et al.*, 2008] A. Procaccia, J. Rosenschein, and A. Zohar. On the complexity of achieving proportional representation. *Social Choice and Welfare*, 30(3):353–362, 2008.

[Skowron and Faliszewski, 2015] P. Skowron and P. Faliszewski. Fully proportional representation with approval ballots: Approximating the MaxCover problem with bounded frequencies in FPT time. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence (AAAI)*, pages 2124–2130, 2015.

[Skowron *et al.*, 2015a] P. Skowron, P. Faliszewski, and A. Slinko. Achieving fully proportional representation: Approximability results. *Artificial Intelligence*, 222:67–103, 2015.

[Skowron *et al.*, 2015b] P. Skowron, L. Yu, P. Faliszewski, and E. Elkind. The complexity of fully proportional representation for single-crossing electorates. *Theoretical Computer Science*, 569:43–57, 2015.

[Williamson and Shmoys, 2011] D. P. Williamson and D. B. Shmoys. *The Design of Approximation Algorithms*. Cambridge University Press, 2011.