Maximizing Nash product social welfare in allocating indivisible goods

Andreas Darmann and Joachim Schauer

Abstract

We consider the problem of allocating indivisible goods to agents who have preferences over the goods. In such a setting, a central task is to maximize social welfare. In this paper, we assume the preferences to be additive, and measure social welfare by means of of the Nash product. We focus on the computational complexity involved in maximizing Nash product social welfare when scores inherent in classical voting procedures such as Approval or Borda voting are used to associate utilities with the agents' preferences. In particular, we show that the maximum Nash product social welfare can be computed efficiently when Approval scores are used, while for Borda and Lexicographic scores the problem becomes NP-complete.

1 Introduction

The allocation of goods (items, resources) to agents who have preferences over these goods (multiagent resource allocation) is a fundamental problem of economics, and, in particular, social choice theory. This problem has been tackled in various scenarios (see, e.g., Chevaleyre et al. (2006) for a survey), where, e.g., we distinguish between divisible and indivisible goods, and centralized and decentralized approaches. Here, we consider the case of indivisible and nonshareable goods to be distributed among agents who report their preferences to a central authority. Typically, individual utilities of (bundles of) items are associated with the preferences over the items. In this work, this is done via numerical scores used in voting rules. Now, a major task is to find an allocation which maximizes the social welfare achieved. Different notions of social welfare have been introduced, the most important being utilitarian, egalitarian, and Nash Product social welfare (cf. Brandt et al. (2013)).
Loosely speaking, utilitarian social welfare of an allocation is given by the sum of the agents' utilities resulting from the allocation. A more fine-grained approach is egalitarian social welfare, where the lowest of the agents' individual utilities in a given allocation is considered. In a certain sense, the Nash product social welfare links these two approaches: by measuring the product of the agents' utilities in an allocation, maximizing the Nash product social welfare targets at a "balanced" allocation (see also Nguyen et al. (2014)). In particular, the Nash product increases when inequality among two agents is reduced (given the respective change is mean-preserving; see also Ramezani & Endriss (2010)). For further desirable properties that are satisfied by the Nash product, such as independence of individual scale of utilities, we refer to Moulin (2003).

A central question in maximizing social welfare is the computational complexity involved. We assume that the agents have additive preferences, i.e., for each agent, the utility of a set of goods is the sum of the utilities of the single goods it contains.
Clearly, maximizing utilitarian social welfare is an easy task – simply allocate each item to an agent who it yields the highest utility for (see also Brandt et al. (2013)). In contrast, it is known that maximizing egalitarian social welfare and Nash Product social welfare are NP-complete for additive utilities and general scoring functions (Roos & Rothe (2010)). Very recently, Baumeister et al. (2013) have shown that maximizing egalitarian social welfare remains NP-complete for a number of prototypical scoring functions: Quasi-Indifference, Borda, and Lexicographic scoring. On the positive side, it is known that the maximum egalitarian social welfare can be computed in polynomial time for Approval scores (Golovin (2005)). To the best of our knowledge, the computational complexity of maximizing Nash product social welfare under scoring functions such as Approval, Borda, or Lexicographic scoring

has not been considered yet. In this paper, we investigate the computational complexity involved in maximizing Nash product social welfare under these classical scoring functions.

Related work and our contribution. In the context of maximizing social welfare in multiagent resource allocation, complexity results have been achieved with respect to different types of utility representation: the bundle form, k-additive form, or straight-line programs. For the bundle form representation, NP-completeness results for utilitarian (Chevaleyre et al. (2008)), egalitarian (Roos & Rothe (2010)), and Nash product social welfare (Roos & Rothe (2010) and Ramezani & Endriss (2010)) are known. For straight-line programs, Dunne et al. (2005) show that maximizing utilitarian social welfare is NP-complete, while Nguyen et al. (2014) show that maximizing social welfare is NP-complete both for the egalitarian and Nash product approach. Both maximizing egalitarian social welfare and maximizing Nash Product social welfare turn out to be NP-complete for 1-additive, i.e., additive utilities already (Lipton et al. (2004) and Roos & Rothe (2010)). In these works, reductions from Partition are given, which do not imply the NP-completeness for any of the scoring functions considered in our work. Given additive utilities, very recently Baumeister et al. (2013), besides many other results, have proven that maximizing egalitarian social welfare is NP-complete for Borda, Lexicographic, and Quasi-Indifference scoring.
In this paper, we show that maximizing Nash Product social welfare is NP-complete for Borda and Lexicographic scores, whereas it is polynomially solvable for Approval scores. The computational complexity involved when Quasi-Indifference scores are used is still open.

2 Formal Framework

2.1 Preliminaries

Let $R = \{r_1, r_2, \ldots, r_m\}$ be a set of m indivisible resources (items) and let $A = \{a_1, \ldots, a_n\}$ be a set of n agents. An *allocation* is a mapping that assigns to each agent a subset of resources such that each resource is handed to exactly one agent. Formally, an allocation P is a mapping $P : A \to 2^R$ with $\bigcup_{a \in A} P(a) = R$ and $P(a_i) \cap P(a_j) = \emptyset$ whenever $i \neq j$.
Now, in our model, we start with ordinal inputs, i.e., the agents rank resources, and map these ranks to numerical scores then. Note that we do not claim that these numerical scores are equivalent or at least close to the agents' actual utilities. However, starting with numerical inputs instead would have several drawbacks (see also Baumeister et al. (2013)); e.g., often it is easier for agents to rank items instead of associating numerical values with each single item, especially in contexts where money is not a key factor. Next, as also pointed out in Baumeister et al. (2013), the use of numerical inputs has the severe disadvantage that it insinuates comparability of interpersonal preferences. Finally, note that our approach is very common in voting theory, as in fact it resembles the way that positional scoring rules proceed[1].
In particular, we assume that agents have preferences over the single resources. The preferences are expressed by means of strict orders $\succ_{a_i}$ over R, which are summarized by the n-tuple $\pi = (\succ_{a_1}, \succ_{a_2}, \ldots, \succ_{a_n})$ called *profile*. We denote by $rank_{a_i}(r)$ the rank of resource r in the ranking of agent a_i.
We adopt scores used in voting procedures to evaluate these preferences by means of utility functions $u_a : R \to \mathbb{Q}$, $a \in A$. We assume that the utility functions are additive, i.e., for any subset $R' \subseteq R$ we have $u_a(R') = \sum_{r \in R'} u_a(R')$. For the sake of readability, we may write $u_a(P)$ instead of $u_a(P(a))$.
Given a profile π, we consider the following types of scores (where $r \in R$):

- *k-approval scores:* For each agent $a \in A$,

$$u_a(r) = \begin{cases} 1 & \text{if } rank_a(r) \leq k \\ 0 & \text{otherwise} \end{cases}$$

[1]Obviously, with the clear difference that we are finally interested in allocations instead of winners of elections.

- *Borda scores:* For each agent $a \in A$, $u_a(r) = m + 1 - rank_a(r)$.
- *Lexicographic scores:* For each agent $a \in A$, $u_a(r) = 2^{m-rank_a(r)}$.

Given k-approval scores, for each $a \in A$, u_a partitions the set R into a set $S_a := \{r \in R : u_a(r) = 1\}$ (the set of resources agent a approves of) and a set $S_a^c := \{r \in R : u_a(r) = 0\}$ (the set of resources agent a disapproves of). Conversely, specifying the set S_a (of size k) for each agent a uniquely determines the corresponding k-approval scores. More generally (and slightly abusing notation), given a set $S(a) \subseteq R$ for each $a \in A$, *Approval scores* are given by $u_a(r) = 1$ for $r \in S(a)$ and $u_a(r) = 0$ for $r \in R \setminus S(a)$.

Given an allocation P, the *Nash product social welfare* for P is given by $sw_N(P) = \prod_{1 \le i \le n} u_{a_i}(P)$.

2.2 Problem Definitions

In this paper, we consider the problem of maximizing the Nash product social welfare with respect to the above scores, i.e., utility functions. The corresponding decision problems are defined as follows.

Definition 2.1 *(*Nash Product Social Welfare Maximization-*approval)*

GIVEN:	*Quadruple (R, A, S, k): R is a set of resources, A a set of agents, a collection $S = \{S_{a_1}, S_{a_2}, \ldots, S_{a_n}\}$ of subsets $S_{a_i} \subseteq R$, and $k \in \mathbb{N}$.*
QUESTION:	*Is there an allocation P such that $sw_N(P) \ge k$, where $u_{a_i}(r) = 1$ if $r \in S_{a_i}$ and $u_{a_i}(r) = 0$ otherwise?*

Analogously, we define Nash Product Social Welfare Maximization-*Borda*.

Definition 2.2 *(*Nash Product Social Welfare Maximization-*Borda)*

GIVEN:	*Quadruple (R, A, π, k): R is a set of resources, A a set of agents, π is a profile, and $k \in \mathbb{N}$.*
QUESTION:	*Is there an allocation P such that $sw_N(P) \ge k$ for Borda scores?*

It is straightforward to define Nash Product Social Welfare Maximization-*lexicographic* for lexicographic scores. In what follows, we use the shortcut NPSW for Nash Product Social Welfare Maximization.

3 Complexity of NPSW

3.1 The easy case: NPSW-Approval

First, we show that NPSW is in P for approval scores. This is done by a transformation to the polynomially solvable Min Cost Flow problem (cf. Ahuja et al. (1993)). We begin with some basic definitions and two known properties of a min cost flow (i.e., an optimal solution of the Min Cost Flow problem).

Definition 3.1 *In an instance $\mathcal{M} = (G, c, \ell, p, b)$ of* Min Cost Flow*, we are given a directed graph $G = (V, E)$. With each edge $e \in E$, two rational numbers are associated: a cost $c(e)$ and an upper bound $p(e)$ on the capacity of e. For each $v \in V$, we are given the rational-valued vertex demand $b(v)$. The* Min Cost Flow *problem can be stated as follows:*

$$\min \sum_{(u,v) \in E} c(u,v) f(u,v)$$

$$\begin{array}{llcll} s.t. & \sum_{v:(u,v)\in E} f(u,v) - \sum_{v:(v,u)\in E} f(u,v) & = & b(u) & \textit{for all } u \in V \\ & 0 \le f(u,v) \le p(u,v) & & & \textit{for all}(u,v) \in E \end{array} \quad (1)$$

A function $f : E \to \mathbb{Q}$ is called flow, if f satisfies the conditions stated in (1). The cost of a flow f is defined by $c(f) = \sum_{(u,v)\in E} c(u,v)f(u,v)$.

In an instance $\mathcal{M} = (G, c, \ell, p, b)$ of Min Cost Flow, the capacity constraints on the edges are written by means of $[0, p(e)]$. The cost of a directed cycle defined as the sum of the costs of the edges in the cycle.
In $\mathcal{M}$, we associate a *residual network* G_f with a flow f. G_f is constructed from G as follows. Each edge $(i, j) \in E$ is replaced by the edges (i, j) and (j, i). In G_f, the arc (i, j) has cost $c(i, j)$ and residual capacity $[0, p(i, j) - f(i, j)]$; the arc (j, i) has cost $c(j, i) = -c(i, j)$ and residual capacity $[0, f(i, j)]$. Finally, G_f consists of edges with positive residual capacity only.

Theorem 3.1 *(Negative cycle optimality condition; cf. Ahuja et al. (1993)) A flow f is an optimal solution of* Min Cost Flow*, if and only if G_f does not contain a negative cost directed cycle.*

Theorem 3.2 *(Integrality property; cf. Ahuja et al. (1993)) If all arc capacities and all node demands are integer, then there is an integer min cost flow.*

Theorem 3.3 *NPSW-Approval is in* P.

Proof. Let $\mathcal{I} = (R, A, \pi, k)$ be an instance of NPSW-Approval. We assume that each item is approved of by at least one agent (otherwise, items with are not approved by any agent are removed in a preprocessing step). We argue that $\mathcal{I}$ can be decided by solving an instance $\mathcal{M}$ of Min Cost Flow. $\mathcal{M}$ is defined as follows. In the graph $G = (V, E)$, certain vertices are identified with items/agents of the same label. In particular, $V = \{s, t\} \cup A \cup R \cup \{t_{i,j} | i \in A, j \in R\}$. The vertex demands are $b(s) = m$, $b(t) = -m$ and $b(v) = 0$ for each $v \in V \setminus \{s, t\}$. In order to construct the edge set E,

- for each $r \in R$ we introduce edge (s, r) with capacity $[0, 1]$ and zero cost.
- for each $a_i \in A$ and for each $r \in R$ with $u_{a_i}(r) = 1$ we introduce the edge (r, a_i) with capacity $[0, 1]$ and zero cost.
- for each $a_i \in A$ and $1 \le j \le m$, we introduce
  - the edge $(a_i, t_{i,j})$ with capacity $[0, 1]$ and cost $c(a_i, t_{i,j}) = n^j$
  - the edge $(t_{i,j}, t)$ with capacity $[0, 1]$ and zero cost.

By the integrality property, there is an integer min cost flow f in $\mathcal{M}$. I.e., for each $e \in E$, f either does not send flow along e or f sends exactly 1 unit of flow along e. Clearly, due to the choice of the vertex demands and the edge capacities $[0, 1]$ of the edges (s, r), for each $r \in R$ there is exactly one unit of flow sent through vertex r. Due to the capacities of the edges (r, a) this means that for each $r \in R$, there is exactly one $a \in A$ such that f sends (one unit of) flow along (r, a). Thus, the mapping $P_f : A \to R$ defined by $r \in P_f(a)$ iff $f(r, a) = 1$ is an allocation in $\mathcal{I}$. On the other hand, it is not hard to see that an allocation P induces an integer flow f^P in $\mathcal{M}$ by

- sending one unit of flow along (s, r) for each $r \in R$
- for each $a_i \in A$ and for each $r \in R$, sending one unit of flow along (r, a_i) iff $r \in P(a_i)$
- sending one unit of flow along $(a_i, t_{i,h})$ and $(t_{i,h}, t)$ for each $1 \le h \le u_{a_i}(P)$

The proof proceeds in three steps.

STEP 1: Let f be an integer flow in instance $\mathcal{M}$, where f_i denotes the amount of flow sent through vertex a_i. We show that the following holds: f is a min cost flow if and only if the two properties

1. for each $a_i \in A$, f sends flow along the arcs $(a_i, t_{i,h})$, for all $1 \leq h \leq f_i$, and
2. there is no sequence $(a_{i_1}, r_{j_1}, a_{i_2}, r_{j_2}, \ldots, r_{j_{\ell-1}}, a_{i_\ell})$ with $f_{i_1} - f_{i_\ell} \geq 2$, such that for all $1 \leq h \leq \ell - 1$ we have (i) $(r_{j_h}, a_{i_{h+1}}) \in E$ and (ii) f sends flow along the arc (r_{j_h}, a_{i_h})

are satisfied. Note that the second property reflects the idea that a more "balanced" and thus cheaper flow cannot be immediately derived from f.

In particular, we show that the above conditions are equivalent to the negative cycle optimality condition. First, note that due to the fact that for each edge (s, r_j) demand and upper bound equal to 1, the residual capacity of the edge is 0. I.e., the edge is not contained in the residual network H. Thus, H does not contain any edge emanating from s. Hence, s cannot be part of any cycle in R.

STEP 1a: Assume that one of the two conditions above are not satisfied. Case I considers the case that the first condition is violated. Case II considers the situation that the first condition holds, but the second is violated.

Case I: For some $a_i \in A$, there is an $1 \leq h \leq f_i$ such that f does not send flow along the edges $(a_i, t_{i,h})$. Then, f must send along an edge $(a_i, t_{i,\ell})$ for some $\ell > f_i$. But then it is easy to see that in the residual network H the cycle $\gamma = (t, t_{i,\ell}, a_i, t_{i,h}, t)$ is a cycle of cost $c(\gamma) = -n^\ell + n^h < 0$ due to $\ell > h$.

Case II: For all $a_i \in A$, f sends one unit of flow along the edges $(a_i, t_{i,h})$, $1 \leq h \leq f_i$. Assume there is a pair (a_i, a_j) with $f_i - f_j \geq 2$ such that (i) $(r, a_j) \in E$ and (ii) f sends flow along the arc (r, a_i). Then, in the residual network H the cycle $\gamma' = (t, t_{i,f_i}, a_i, r, a_j, t_{j,f_j+1}, t)$ has negative cost: $c(\gamma') = -n^{f_i} + n^{f_j+1} < 0$, because $f_i - f_j \geq 2$ by assumption.

Hence, if one of the two conditions is violated, there is a negative cost cycle.

STEP 1b: On the other hand, assume H contains a negative cost cycle γ. We show that this implies that at least one of the two conditions is violated. Clearly, s cannot be contained in γ. In addition, γ cannot be made up of vertex t and vertices $t_{i,j}$ only, since each edge $(t_{i,j}, t)$ is of zero cost.

Assume γ does not contain a vertex r_j, $1 \leq j \leq m$. Then, for some i, x, y, $\gamma = (t, t_{i,x}, a_i, t_{i,y}, t)$ holds. Note that $c(\gamma) = -c(t_{i,x}, a_i) + c(a_i, t_{i,y}) = -n^x + n^y$. Thus, $c(\gamma) < 0$ implies

$$x > y \tag{2}$$

Assume that f sends flow along $(a_i, t_{i,h})$ for all $1 \leq h \leq f_i$. Then the residual network H must contain (i) the edges $(t_{i,h}, a_i)$ for $1 \leq h \leq f_i$ as only edges with head a_i. Thus, $x \leq f_i$ follows. In addition, since f is integer and in the original network G the upper bound of the capacity of each of the edges $(a_i, t_{i,h})$ equals 1, it follows that f sends exactly one unit of flow along each of the arcs $(a_i, t_{i,h})$, for $1 \leq h \leq f_i$. Hence, H cannot contain any of these edges. Thus, $y > f_i$ must hold. Putting things together, we get $x \leq f_i < y$, in contradiction with (2). Hence, the first condition is violated.

Thus, γ contains a vertex $r \in R$. Assume the first condition is not violated (otherwise there is nothing to show). Then, there is a sequence $(a_{i_1}, r_{j_1}, a_{i_2}, r_{j_2}, \ldots, r_{j_{\ell-1}}, a_{i_\ell})$ such that

$$\gamma = (t, t_{i_1,x}, a_{i_1}, r_{j_1}, a_{i_2}, r_{j_2}, \ldots, r_{j_{\ell-1}}, a_{i_\ell}, t_{i_\ell,y}, t)$$

for some $\ell \geq 2$ and some x, y, with $i_1 \neq i_\ell$.

Since by assumption the first condition is satisfied, $x \leq f_{i_1}$ and $y \geq f_{i_\ell} + 1$ hold. $c(\gamma) = -c(t_{i_1,x}, a_{i_1}) + c(a_{i_\ell}, t_{i_\ell,y}) = -n^x + n^y \geq -n^x + n^{f_{i_\ell}+1}$. Now, $c(\gamma) < 0$ implies $-n^x + n^{f_{i_\ell}+1} < 0$, i.e., $x > f_{i_\ell} + 1$. Hence, $f_{i_1} \geq x > f_{i_\ell} + 1$ holds. Thus, $f_{i_1} \geq f_{i_\ell} + 2$ holds, since all flow values are integer. I.e., the second condition is violated.

As a consequence, the negative cycle condition is in fact equivalent to the two above stated conditions.

STEP 2: Let P' be an allocation that maximizes Nash product social welfare. Throughout this proof, let g be the integer flow induced by allocation P'. We show that g is a min cost flow.
For each $a_i \in A$, g sends $g_i = u_{a_i}(P')$ units of flow through vertex a_i and one unit of flow through each of the arcs $(a_i, t_{i,h})$ for $1 \leq h \leq g_i$. Assume there is a pair (a_i, a_j) with $g_i - g_j \geq 2$ such that (i) $(r, a_j) \in E$ and (ii) g sends flow along the arc (r, a_i). Then, both a_i, a_j approve of item r. Consider the assignment P'' defined by $P''(a) = P(a)$ for $a \in A \setminus \{a_i, a_j\}$, $P''(a_i) = P'(a_i) \setminus \{r\}$ and $P''(a_j) = P'(a_j) \cup \{r\}$. Then,

$$\frac{\prod_{a_i \in A} u_{a_i}(P'')}{\prod_{a_i \in A} u_{a_i}(P')} = \frac{(u_{a_i}(P') - 1)(u_{a_j}(P') + 1)}{u_{a_i}(P')(u_{a_j}(P')} = \frac{u_{a_i}(P')u_{a_j}(P') + u_{a_i}(P') - u_{a_j}(P') - 1}{u_{a_i}(P')(u_{a_j}(P')} > 1$$

where the last inequality follows from $u_{a_i}(P') - u_{a_j}(P') - 1 = g_i - g_j - 1 \geq 1$. This contradicts with the fact that P' maximizes Nash social welfare. With Step 1, if follows that g is a min cost flow.

STEP 3: Let f be an integer min cost flow. We show that $\prod_{a_i \in A} f_i = \prod_{a_i \in A} g_i$ holds. W.l.o.g., we assume $f_1 \geq f_2 \geq \ldots \geq f_n$. Clearly, there is a permutation $\pi : A \to A$ such that $g_{\pi(1)} \geq g_{\pi(2)} \geq \ldots \geq g_{\pi(n)}$ holds. We show that $f_i = g_{\pi(i)}$ for each $1 \leq i \leq n$.
Assume the opposite, i.e., there is an index $k \geq 1$ such that $f_i = g_{\pi(i)}$ for $i < k$ and $f_k \neq g_{\pi(k)}$. If $f_k > g_{\pi(k)}$, then

$$\begin{aligned}
c(f) - c(g) &= \sum_{i=1}^{n} \left(\sum_{h=1}^{f(i)} n^h - \sum_{h'=1}^{g_{\pi(i)}} n^{h'}\right) \\
&= (n + n^2 + \ldots + n^{f_k}) - (n + n^2 + \ldots + n^{g_{\pi(k)}}) + \sum_{i=k+1}^{n} \left(\sum_{h=1}^{f_i} n^h - \sum_{h'=1}^{g_{\pi(i)}} n^{h'}\right) \\
&= n^{g_{\pi(k)}+1} + n^{g_{\pi(k)}+2} \ldots + n^{f_k} + \sum_{i=k+1}^{n} \left(\sum_{h=1}^{f_i} n^h - \sum_{h'=1}^{g_{\pi(i)}} n^{h'}\right) \qquad (3)
\end{aligned}$$

Note that for any fixed $h \in \mathbb{N}$, $\sum_{i=1}^{h} n^h = \frac{n^{h+1}-1}{n-1} - 1$ holds. Thus,

$$\sum_{i=k+1}^{n} \sum_{h'=1}^{g_{\pi(i)}} n^{h'} \leq (n-k) \sum_{h'=1}^{g_{\pi(k)}} n^{h'} < (n-k) \frac{n^{g_{\pi(k)}+1}}{n-1} \leq n^{g_{\pi(k)}+1} \qquad (4)$$

With (4) and $f_k > g_{\pi(k)}$, we get

$$\sum_{i=k+1}^{n} \left(\sum_{h=1}^{f_i} n^h - \sum_{h'=1}^{g_{\pi(i)}} n^{h'}\right) > - \sum_{i=k+1}^{n} \sum_{h'=1}^{g_{\pi(i)}} n^{h'} > -n^{g_{\pi(k)}+1} \geq -n^{f_k}$$

Together with (3) we get $c(f) - c(g) > 0$, in contradiction with the fact that f is a min cost flow.
Analogously, $f_k < g_{\pi(k)}$ leads to a contradiction with the fact that g is an integer min cost flow (because of $c(f) = c(g)$). Thus, $f_i = g_{\pi(i)}$ holds for all $1 \leq i \leq k$. Hence, $\prod_{a_i \in A} f_i = \prod_{a_i \in A} g_i$ follows.

Since g is an integer flow of minimum total cost (step 2), from step 3 $\prod_{a_i \in A} f_i = \prod_{a_i \in A} g_i$ follows for any integer min cost flow f. Hence, in order to maximize the Nash product social welfare, it is sufficient to find an integer min cost flow in instance $\mathcal{M}$. This can be done in polynomial time (see, e.g., Ahuja et al. (1993)). □

3.2 The hard cases: NPSW-Borda and NPSW-Lexicographic

Theorem 3.4 *NPSW-Borda is* NP*-complete.*

Proof. The proof proceeds by a reduction from CUBIC MONOTONE 1-IN-3 SAT (cf. Moore & Robson (2001)) and is omitted here. $\square$

Theorem 3.5 *NPSW-Lexicographic is* NP*-complete.*

Proof. We provide a reduction from the NP-complete problem CUBIC MONOTONE 1-IN-3 SAT (cf. Moore & Robson (2001)). An instance $\mathcal{I} = (X, C)$ of that problem consists of a set of variables X and a set C of clauses over X, such that each clause is made up of exactly three variables of X and each variable occurs in exactly three clauses. In CUBIC MONOTONE 1-IN-3 SAT we ask if there is a truth assignment for X such that exactly one variable is true in each clause of C.
Note that there are no negated literals contained in any clause of C. In addition, observe that $|X| = |C|$ holds. Further note that ϕ can be a satisfying truth assignment in instance $\mathcal{I}$ only if it the number of variables set true under ϕ is exactly $\frac{|X|}{3}$. Thus, $|X|$ is a multiple of 3.
Given an instance $\mathcal{I} = (X, C)$ of CUBIC MONOTONE 1-IN-3 SAT we construct an instance $\mathcal{L} = (R, A, \pi, k)$ of NPSW-Lexicographic as follows. Let $n = |X| = |C|$, and $\ell = 12n$.
R consists of $\ell + 6n + \frac{n}{3}$ items:

- the items $d_1, d_2, \ldots, d_\ell$
- the item sets $X = \{x_1, x_2, \ldots, x_n\}$, and $Y = \{x_{1,1}, x_{1,2}, x_{1,3}, x_{2,1}, x_{2,2}, x_{2,3}, \ldots, x_{n,1}, x_{n,2}, x_{n,3}\}$
- the item sets $B = \{b_{1,1}, b_{1,2}, b_{2,1}, b_{2,2}, \ldots, b_{n,1}, b_{n,2}\}$ and $H = \{h_1, h_2, \ldots, h_{\frac{n}{3}}\}$

Over the sets B, H, X resp. Y we define the rankings τ_B, τ_H, τ_X, and τ_Y as follows:

- $\tau_X = x_1 \succ x_2 \succ \ldots \succ x_n$, $\tau_H = h_1 \succ h_2 \succ \ldots \succ h_{\frac{n}{3}}$,
- $\tau_B = b_{1,1} \succ b_{1,2} \succ b_{2,1} \succ \ldots \succ b_{n,2}$, and $\tau_Y = x_{1,1} \succ x_{1,2} \succ x_{1,3} \succ x_{2,1} \succ \ldots \succ x_{n,3}$

Let $S \in \{B, H, X, Y\}$. For any subset Z of S, by τ_Z we denote the ranking τ_S restricted to the subset Z. Within this proof, we represent the clause $C_i = (x_{i_1,k_1} \vee x_{i_2,k_2} \vee x_{i_3,k_3})$, where $k_j \in \{1, 2, 3\}$ denotes the k_j-th occurrence of variable x_{i_j} in C, by the set $C_i = \{x_{i_1,k_1} \vee x_{i_2,k_2} \vee x_{i_3,k_3}\}$.
A consists of $\ell + 5n$ agents:

- For each $1 \leq i \leq \ell$, the ranking of agent D_i is given by
$$d_1 \succ d_2 \succ \ldots \succ d_\ell \succ \tau_B \succ \tau_Y \succ \tau_X \succ \tau_H$$
- For $1 \leq i \leq n$, the ranking of agent H_i is given by
$$\tau_H \succ x_i \succ d_1 \succ d_2 \succ \ldots \succ d_{5n-1} \succ \tau_B \succ \tau_Y \succ \tau_{X\setminus\{x_i\}} \succ d_{5n} \succ \ldots \succ d_\ell$$
- For $1 \leq i \leq n$, the ranking of agent X_i is given by
$$x_i \succ d_1 \succ \ldots \succ d_n \succ x_{i,1} \succ x_{i,2} \succ x_{i,3} \succ d_{n+1} \succ \ldots \succ d_{5n-6} \succ \\ \succ \tau_B \succ \tau_{Y\setminus\{x_{i,1},x_{i,2},x_{i,3}\}} \succ \tau_{X\setminus\{x_i\}} \succ d_{5n-5} \succ \ldots \succ d_\ell$$
- For $1 \leq i \leq n$, the rankings of the agents $\alpha_i, \beta_i, \gamma_i$ are as follows. Let
$$\tau_{\alpha_i} = x_{i_1,k_1} \succ d_{2n+5} \succ d_{2n+6} \succ \ldots \succ d_{3n+4} \succ x_{i_2,k_2} \succ d_{3n+5} \succ d_{3n+6} \succ \ldots \succ d_{4n+4} \succ x_{i_3,k_3}$$
$$\tau_{\beta_i} = x_{i_2,k_2} \succ d_{2n+5} \succ d_{2n+6} \succ \ldots \succ d_{3n+4} \succ x_{i_3,k_3} \succ d_{3n+5} \succ d_{3n+6} \succ \ldots \succ d_{4n+4} \succ x_{i_1,k_1}$$
$$\tau_{\alpha_i} = x_{i_3,k_3} \succ d_{2n+5} \succ d_{2n+6} \succ \ldots \succ d_{3n+4} \succ x_{i_1,k_1} \succ d_{3n+5} \succ d_{3n+6} \succ \ldots \succ d_{4n+4} \succ x_{i_2,k_2}$$

The ranking of α_i is given by

$$d_1 \succ \ldots \succ d_{n+4} \succ b_{i,1} \succ b_{i,2} \succ d_{n+5} \succ \ldots d_{2n+4} \succ \tau_{\alpha_i} \succ d_{4n+5} \succ \ldots \succ d_{5n-1} \succ \\ \tau_{B \setminus \{b_{i,1}, b_{i,2}\}} \succ \tau_{Y \setminus C_i} \succ \tau_X \succ \tau_H \succ d_{5n} \succ \ldots \succ d_\ell$$

The ranking of β_i (resp. γ_i) results from π_{α_i} by replacing τ_{α_i} with τ_{β_i} (resp. τ_{γ_i}).

Let $M := |R| - 1$, and

$$\kappa = [2^{M-(n+1)} + 2^{M-(n+2)} + 2^{M-(n+3)}]^{\frac{2n}{3}} \cdot [2^{M-(n+4)} \cdot 2^{M-(n+5)} \cdot 2^{M-(3n+6)}]^{\frac{n}{3}}$$

Set

$$k = (\prod_{i=0}^{\ell-1} 2^{M-i}) \cdot 2^{\frac{n}{3}} \cdot 2^{\sum_{i=0}^{\frac{n}{3}-1}(M-i)} \cdot 2^{(M-\frac{n}{3})\frac{2n}{3}} \cdot \kappa$$

In instance $\mathcal{L}$ we ask if there is an allocation P with $sw_N(P) \geq k$.

We begin with two simple lemmata.

Lemma 3.6 *Let P be an allocation. Let Q result from P by handing an item $p \in P(a_2)$ to agent a_1 such that the following properties are satisfied:*

- *$\exists q \in P(a_2)$ such that a_2 ranks q higher than p*
- *a_1 ranks p higher than the highest-ranked item of $P(a_1)$*

Then, $\prod_{a \in A} u_a(P) < \prod_{a \in A} u_a(Q)$.

Proof. Let $u_{a_2}(p) = 2^s$ and $u_{a_1}(p) = 2^t$ for some $s, t \in \mathbb{N}$. Clearly, the stated properties imply

$$u_{a_2}(P) > 2^{s+1} \text{ and } u_{a_1}(P) < 2^t \tag{5}$$

With (5), we can conclude that

$$\begin{aligned} u_{a_1}(Q) \cdot u_{a_2}(Q) &= [u_{a_1}(P) + 2^t)] \cdot [u_{a_2}(P) - 2^s] \\ &= u_{a_1}(P) \cdot u_{a_2}(P) + 2^t u_{a_2}(P) - 2^s u_{a_1}(P) - 2^{s+t} \\ &> u_{a_1}(P) \cdot u_{a_2}(P) + 2^{t+s+1} - 2^s 2^t - 2^{s+t} \\ &= u_{a_1}(P) \cdot u_{a_2}(P) \end{aligned}$$

holds. Therefore, $\prod_{a \in A} u_a(P) < \prod_{a \in A} u_a(Q)$. ◊

Lemma 3.7 *Let P be an allocation. Let Q result from P by handing an item $p \in P(a_2)$ to agent a_1 and $q \in P(a_1)$ to agent a_2 such that for some $j \in \mathbb{N}$ the following properties are satisfied:*

- *a_2 ranks q at most j positions lower than p*
- *a_1 ranks p at least $(j+1)$ positions higher than the highest-ranked item of $P(a_1)$*

Then, $\prod_{a \in A} u_a(P) < \prod_{a \in A} u_a(Q)$.

Proof. Let λ denote the rank of the highest-ranked item of $P(a_1)$ in the ranking of a_1. Then,

$$2^\lambda \leq u_{a_1}(P) < 2^{\lambda+1} \tag{6}$$

holds. Let μ denote the rank of item p in the ranking of a_2; $2^\mu \leq u_{a_2}(P)$ holds. Comparing the Nash product social welfare achieved by the allocations, it is enough to consider

$$u_{a_1}(Q) \cdot u_{a_2}(Q) - u_{a_1}(P) \cdot u_{a_2}(P)$$

since for the remaining the agents the utilities of P and P' coincide. Now,

$$\begin{aligned}
&u_{a_1}(Q) \cdot u_{a_2}(Q) - u_{a_1}(P) \cdot u_{a_2}(P) \\
&\quad \geq \quad (u_{a_1}(P) + 2^{\lambda+(j+1)} - 2^\lambda)(u_{a_2}(P) + 2^{\mu-j} - 2^\mu) - u_{a_1}(P) \cdot u_{a_2}(P) \\
&\quad = \quad u_{a_1}(P)(2^{\mu-j} - 2^\mu) + (2^{\lambda+(j+1)} - 2^\lambda)u_{a_2}(P) + 2^{\lambda+\mu+1} - 2^{\lambda+(j+1)+\mu} - 2^{\lambda+\mu-j} + 2^{\lambda+\mu} \\
&\quad = \quad [u_{a_1}(P)2^{\mu-j} - 2^{\lambda+\mu-j}] + [2^{\lambda+\mu+1} - 2^\mu u_{a_1}(P)] + [(2^{\lambda+(j+1)} - 2^\lambda)u_{a_2}(P) - 2^{\lambda+(j+1)+\mu} + 2^{\lambda+\mu}] \\
&\quad > \quad [(2^{\lambda+(j+1)} - 2^\lambda)u_{a_2}(P) - 2^{\lambda+(j+1)+\mu} + 2^{\lambda+\mu}]
\end{aligned}$$

where the last inequality follows from (6). Since $u_{a_2}(P) \geq 2^\mu$, we hence get

$$u_{a_1(Q)\cdot u_{a_2}(Q)-u_{a_1}(P)\cdot u_{a_2}(P)} > [(2^{\lambda+(j+1)} - 2^\lambda)2^\mu - 2^{\lambda+(j+1)+\mu} + 2^{\lambda+\mu}] = 0$$

As a consequence, $\prod_{a\in A} u_a(P) < \prod_{a\in A} u_a(Q)$. $\Diamond$

Claim. $\mathcal{I}$ is a "yes"-instance of Cubic Monotone 1-in-3 Sat if and only if $\mathcal{L}$ is a "yes"-instance of NPSW-Lexicographic.

Proof of Claim. *"Only-if"-part:* Let ϕ be a truth assignment that sets true exactly one variable in each clause. Abusing notation, we identify ϕ with the set of variables set true under ϕ. Recall that $|\phi| = \frac{n}{3}$. We define the allocation P as follows.

- $P(D_i) = d_i$ for each $1 \leq i \leq \ell$. Thus, the total product of the utilities of these agents is $\prod_{i=0}^{\ell-1} 2^{M-i}$.
- For each $x_i \in \phi$, let $P(X_i) = x_i$,
  - for the q-th clause C_j that contains x_i, $q \in \{1, 2, 3\}$, assign $x_{i,q}$ to the one among α_j, β_j, γ_j that ranks $x_{i,q}$ highest (i.e., directly below d_{2n+4}, in position $3n+7$); for the two remaining agents among α_j, β_j, γ_j, allocate $b_{i,1}$ to one and $b_{i,2}$ to the other agent.
  - allocate exactly one of $\{h_1, \ldots, h_{\frac{n}{3}}\}$ to agent H_i.

  Hence, the total product of the utilities of these agents is

  $$(2^M)^{\frac{n}{3}} \cdot (2^{M-(3n+6)} \cdot 2^{M-(n+4)} \cdot 2^{M-(n+5)})^{\frac{n}{3}} \cdot (2^M \cdot 2^{M-1} \cdots 2^{M-\frac{n}{3}+1})$$

- For each $x_i \notin \phi$, let $P(H_i) = x_i$ and $P(X_i) = \{x_{i,1}, x_{i,2}, x_{i,3}\}$. The total product of the utilities of these agents is
  $$(2^M)^{\frac{2n}{3}} \cdot (2^{M-(n+1)} + 2^{M-(n+2)} + 2^{M-(n+3)})^{\frac{2n}{3}}$$

Thus, $\prod_{a\in A} u_a(P) = k$, implying that $\mathcal{L}$ is a "yes"-instance.

"If"-part: Let P be an allocation with $\prod_{a\in A} u_a(P) \geq k$. This implies that the maximum Nash product social welfare achieved exceeds the threshold k. W.l.o.g. we assume that P is an allocation of maximum Nash product social welfare. We show that P must satisfy several properties:

1. ***d_i is allocated to D_i for each $1 \leq i \leq n$:*** Assume there is an agent D_i such that $P(D_i) \cap \{d_1, \ldots, d_\ell\} = \emptyset$. Let $r \in P(D_i)$ denote the item which D_i ranks highest among the items in $P(D_i)$. We distinguish the following cases.

(a) There is a D_j who gets allocated at least two elements of $\{d_1, \dots, d_\ell\}$. Let $d_{\min}$ be the lowest ranked of these items in the ranking of D_j. Consider the allocation P' which results from P by handing $d_{\min}$ to D_i. With Lemma 3.6, $\prod_{a \in A} u_a(P) < \prod_{a \in A} u_a(P')$ holds which contradicts with the choice of P.

(b) There is an agent $a \neq D_j$ who gets allocated at least one of $\{d_1, \dots, d_\ell\}$. Take an arbitrary such $d \in P(a)$. Consider the allocation P'' which results from P by handing r to a and d to D_i. If a ranks r above d, trivially $u_a(P'') > u_a(P)$ and $u_{D_i}(P'') > u_{D_i}(P)$ follow, since D_i by construction ranks d above r.
Assume a ranks r below d. We can observe that in the ranking of D_i, r is among the last $(6n + \frac{n}{3})$ positions. For any other agent, r is ranked higher by construction. Thus, a ranks r higher than D_i does; also by construction, D_i ranks d at least as high as a does. Hence, the number μ of items between d and r in the ranking of D_i exceeds the number of items between d and r in the ranking of a by at least one item. In other words, Lemma 3.7 can be applied, again leading to a contradiction with the choice of P.

As a consequence, each agent D_i gets at least (and thus exactly) one item of $\{d_1, \dots d_\ell\}$. Since the rankings of the agents D_i, $1 \leq i \leq \ell$, coincide, w.l.o.g. we assume that d_i is allocated to D_i.

2. *h_i is allocated to one of $\{H_1, \dots, H_n\}$, for each $1 \leq i \leq \frac{n}{3}$.* Assume h_i is allocated to an agent $a \notin \{H_1, \dots, H_n\}$. Then, take an arbitrary $H_j \in \{H_1, \dots, H_n\}$ who is not allocated any item of $\{h_1, \dots, h_{\frac{n}{3}}\}$. Obviously, such an agent H_j exists. It is easy to see that for the allocation $\bar{P}$ which results from P by handing h_i to H_j, and, in turn, any item of $P(H_j)$ to a satisfies $u_{H_j}(\bar{P}) > u_{H_j}(P)$ and $u_a(\bar{P}) > u_a(P)$, i.e., $\prod_{a \in A} u_a(P) < \prod_{a \in A} u_a(\bar{P})$.

3. *x_i is allocated to one of $\{H_i, X_i\}$, for each $1 \leq i \leq n$.* Assume x_i is assigned to an agent $a \notin \{H_i, X_i\}$. Consider the allocation $\tilde{P}$ which results from P by handing x_i to X_i, and, in turn, the item r' of $P(X_i)$ which X_i ranks highest to agent a. Recall that $r' \notin \{d_1, \dots, d_\ell, h_1, \dots h_{\frac{n}{3}}\}$. If $r' \notin \{x_{i+1}, \dots, x_n\}$, then obviously $u_{X_i}(\tilde{P}) > u_{X_i}(P)$ and $u_a(\tilde{P}) > u_a(P)$ hold, i.e., $\prod_{a \in A} u_a(P) < \prod_{a \in A} u_a(\tilde{P})$. Let $r' \in \{x_{i+1}, \dots, x_n\}$. Then, X_i ranks x_i more than $3n$ positions above r'. Note that any agent $a \notin \{H_i, X_i\}$ ranks r' at most $(n-1)$ positions below x_i. Thus, the conditions stated in Lemma 3.7 are satisfied, and again we get a contradiction with the choice of P.

4. *H_j is allocated exactly one of $\{h_1, \dots, h_{\frac{n}{3}}\} \cup \{x_j\}$, for each $1 \leq j \leq n$.* Note that with Step 3 this means that H_j is allocated exactly one of $\{h_1, \dots, h_{\frac{n}{3}}\} \cup \{x_1, \dots, x_n\}$. This step is split in three parts:

(a) *H_j is allocated at most one of $\{h_1, \dots, h_{\frac{n}{3}}\}$, for each $1 \leq j \leq n$.* Assume there is an agent H_j who is allocated at least two items of $\{h_1, \dots, h_{\frac{n}{3}}\}$. Let h_g be the lower-ranked of the two items in the ranking of H_j. For the allocation $\hat{P}$ which results from P by handing h_g to an agent $H_{j'}$ who P does not allocate an item of $\{h_1, \dots, h_{\frac{n}{3}}\}$ to, we get with Lemma 3.6 that $\prod_{a \in A} u_a(P) < \prod_{a \in A} u_a(\hat{P})$ holds.

(b) *If H_j is allocated one of $\{h_1, \dots, h_{\frac{n}{3}}\}$, then H_j is not allocated x_j.* Let $h \in \{h_1, \dots, h_{\frac{n}{3}}\}$ be allocated to H_j. Assume the opposite. Consider the allocation Q which results from P by handing x_j to X_j. Again, with Lemma 3.6 $u_{H_j}(Q) \cdot u_{X_i}(Q) > u_{H_j}(P) \cdot u_{X_i}(P)$ follows.

(c) *If H_j is allocated none of $\{h_1, \dots, h_{\frac{n}{3}}\}$, then H_j is allocated x_j.* Assume the opposite. Since by assumption $\prod_{a \in A} u_a(P) > 0$, P must allocate an item r to agent H_j. Again, let r be the item highest-ranked by H_j that H_j receives under P.
From Steps 1-3, we can conclude that $r \in \{b_{1,1}, \dots, b_{n,2}\} \cup \{x_{1,1}, \dots, x_{n,3}\}$ holds. Consider the allocation Q' which results from P by handing x_j to H_j, and, in turn, item r to X_j. By construction, X_i ranks r more than $\frac{n}{3}$ positions higher than H_j does. On the other hand, H_j ranks x_j exactly $\frac{n}{3}$ positions lower than X_i does. Therefore, the number μ of items

between x_j and r in the ranking of H_j exceeds the number of items between x_j and r in the ranking of X_i by at least one item. As a consequence, Lemma 3.7 yields a contradiction with the choice of P.

5. *Two of $\{\alpha_i, \beta_i, \gamma_i\}$ are allocated exactly one of $\{b_{i,1}, b_{i,2}\}$, for each $1 \leq i \leq n$.* This step is proven in two parts.

   (a) *$b_{i,1}$ (resp. $b_{i,2}$) is allocated to α_i, β_i or γ_i , for each $1 \leq i \leq n$.* Assume $b_{i,1}$ is not allocated to one of these agents. Clearly, at most one of α_i, β_i, γ_i is allocated $b_{i,2}$. W.l.o.g. assume $b_{i,2}$ is not allocated to α_i. By Step (1), this implies that α_i ranks $b_{i,1}$ more than $2n$ positions higher than the highest-ranked among the items in $P(\alpha_i)$. Take an arbitrary $p \in P(\alpha_i)$. Note that with Steps (1)-3, $p \notin \{d_1, \ldots, d_\ell, x_1, \ldots, x_n, h_1, \ldots, h_{\frac{n}{3}}\} \cup \{b_{i,2}\}$ follows. Consider the allocation Q which results from P by handing p to the agent a with $b_{i,1} \in P(a)$ and, in turn, $b_{i,1}$ to agent α_i. By construction (in particular, by the items $d_{n+5}, \ldots, d_{2n+4}$ in the ranking of α_i), it follows that the conditions of Lemma 3.7 are satisfied. Thus $\prod_{a\in A} u_a(P) < \prod_{a\in A} u_a(Q)$ holds, in contradiction with the choice of P.

   (b) *$b_{i,1}$ and $b_{i,2}$ are not allocated to the same agent, for each $1 \leq i \leq n$.* Assume the opposite. Then, analogously to above, by the use of Lemma 3.7 we can find an allocation with a higher Nash product social welfare than P.

6. *For each $1 \leq i \leq n$ and $a \in \{\alpha_i, \beta_i, \gamma_i\}$, the following holds: If $b_{i,1}$ or $b_{i,2}$ is allocated to a, then a is allocated no further item.* It remains to show that no item of Y is allocated to a. We provide a proof for agent α_i and $b_{i,2} \in P(\alpha_i)$ (the other cases follow analogously). Assume at least one of Y is allocated to α_i. Let $x_{g,j}$ be allocated to α_i. Consider the allocation Q' which results from P by handing $x_{g,j}$ to agent X_g. Let $u_{\alpha_i}(x_{g,j}) = 2^\varepsilon$ for some $\varepsilon \in \mathbb{N}$. Note that $u_{X_g}(x_{g,j}) \geq 2^{\varepsilon+2n}$ and
$$u_{\alpha_i}(P) \geq 2^{M-(n+4)} + 2^\varepsilon \tag{7}$$
hold. We get
$$\begin{aligned}
& u_{X_g}(Q') \cdot u_{\alpha_i}(Q') - u_{X_g}(P) \cdot u_{\alpha_i}(P) \\
\geq\ & (u_{X_g}(P) + 2^{\varepsilon+2n}) \cdot (u_{\alpha_i}(P) - 2^\varepsilon) - u_{X_g}(P) \cdot u_{\alpha_i}(P) \\
\geq\ & [u_{X_g}(P) \cdot u_{\alpha_i}(P) - 2^\varepsilon u_{X_g}(P) + 2^{\varepsilon+2n} 2^{M-(n+4)}] - u_{X_g}(P) \cdot u_{\alpha_i}(P) \\
>\ & -2^{\varepsilon+M+1} + 2^{\varepsilon+M+n-4} \\
>\ & 0
\end{aligned}$$
where the third line follows from (7), the fourth from $u_{X_g}(P) < 2^{M+1}$, and the last from $n > 5$.

7. *For each $1 \leq i \leq n$ and $a \in \{\alpha_i, \beta_i, \gamma_i\}$, the following holds: If a is allocated an item of $\{x_{i_1,k_1}, x_{i_2,k_2}, x_{i_3,k_3}\}$, then a is allocated exactly one item.* This follows analogously to Step 6.

8. *None of the items in Y is allocated to an agent $a \in \{D_1, \ldots, D_\ell, H_1, \ldots, H_n\}$.* Assume the opposite. Take an arbitrary item $x_{g,j} \in P(a) \cap Y$. Note that $u_a(x_{g,j}) = 2^\varepsilon$ and $u_{X_g}(x_{g,j}) \geq 2^{\varepsilon+3n}$ for some $\varepsilon < M - \ell$. Thus,
$$u_a(P) \geq 2^\varepsilon + 2^{M-\frac{n}{3}} \tag{8}$$
Consider the allocation Q'' which results from P by handing $x_{g,j}$ to agent X_g. With (8),
$$\begin{aligned}
& u_{X_g}(Q'') \cdot u_a(Q'') - u_{X_g}(P) \cdot u_a(P) \\
\geq\ & (u_{X_g}(P) + 2^{\varepsilon+3n}) \cdot (u_a(P) - 2^\varepsilon) - u_{X_g}(P) \cdot u_a(P) \\
>\ & [u_{X_g}(P) \cdot u_a(P) - 2^\varepsilon u_{X_g}(P) + 2^{\varepsilon+3n} 2^{M-\frac{n}{3}} - u_{X_g}(P) \cdot u_a(P) \\
>\ & -2^{\varepsilon+M+1} + 2^{\varepsilon+M+2n} \\
>\ & 0
\end{aligned}$$

and thus a contradiction with the choice of P is implied.

As an immediate consequence, we know that (i) each of $D_1, \ldots, D_\ell, H_1, \ldots, H_n$ is allocated exactly one item (follows from Steps 1-5 and Step 8), and (ii) there are at most $2n$ items available for the agents $X_1, \ldots, X_n$ (by the pigeonhole principle), all of which belonging to the set Y. From (i), it follows with steps 1 and 4 that

$$\prod_{a \in \{D_1, \ldots, D_\ell\}} u_a(P) \cdot \prod_{a \in \{H_1, \ldots, H_n\}} u_a(P) = (\prod_{i=0}^{\ell-1} 2^{M-i}) \cdot 2^{\sum_{i=0}^{\frac{n}{3}-1}(M-i)} \cdot 2^{(M-\frac{n}{3})\frac{2n}{3}} \tag{9}$$

From Steps 3 and 4 we know that there are exactly $\frac{n}{3}$ agents among the agents X_i that are allocated x_i, $1 \leq i \leq n$, while the remaining $\frac{2n}{3}$ agents among $X_1, \ldots, X_n$ are not allocated any item of X. Keeping in mind that the Nash product is maximized for the most balanced allocation, it is not difficult to verify that the following observation holds.

Observation. *If at most $2n$ items of Y are contained in $\bigcup_{a \in \{X_1, \ldots, X_n\}} P(a)$, then*

$$\prod_{a \in \{X_1, \ldots, X_n\}} u_a(P) \leq (2^M)^{\frac{n}{3}} \cdot (2^{M-(n+1)} + 2^{M-(n+2)} + 2^{M-(n+3)})^{\frac{2n}{3}} \tag{10}$$

holds; equality in (10) is achieved if and only if for all $1 \leq i \leq n$, all the items $\{x_{i,1}, x_{i,2}, x_{i_3}\}$ are allocated to the agent X_i satisfying $x_i \notin P(X_i)$.

Now, assume for some i, there is an $a \in \{\alpha_i, \beta_i, \gamma_i\}$ which is allocated none of $\{b_{i,1}, b_{i,2}\} \cup \{x_{i_1,k_1}, x_{i_2,k_2}, x_{i_3,k_3}\}$. With Step 1, $u_a(P) < 2^{M-7n}$ follows. With Steps 6 and 7, we get

$$\prod_{a \in \{\alpha_i, \beta_i, \gamma_i | 1 \leq i \leq n\}} u_a(P) < (2^{M-7n} \cdot 2^{M-(n+4)} \cdot 2^{M-(n+5)}) \cdot (2^{M-(3n+6)} \cdot 2^{M-(n+4)} \cdot 2^{M-(n+5)})^{n-1} \tag{11}$$

Combining (9), (10), (11), $\prod_{a \in A} < k$ follows.
Thus, for each i and $a \in \{\alpha_i, \beta_i, \gamma_i\}$, a is allocated at least – by Steps 6 and 7, that means exactly – one of $\{b_{i,1}, b_{i,2}\} \cup \{x_{i_1,k_1}, x_{i_2,k_2}, x_{i_3,k_3}\}$. With Step 5, we can conclude that exactly one agent of $\{\alpha_i, \beta_i, \gamma_i\}$ is allocated exactly one of $\{x_{i_1,k_1}, x_{i_2,k_2}, x_{i_3,k_3}\}$, obviously yielding an utility of $2^{M-(3n+6)}$. Hence, we get

$$\prod_{a \in \{\alpha_i, \beta_i, \gamma_i | 1 \leq i \leq n\}} u_a(P) = (2^{M-(3n+6)} \cdot 2^{M-(n+4)} \cdot 2^{M-(n+5)})^n \tag{12}$$

With (9) and (12), the above observation implies that

- for each clause C_i, exactly one of $\{x_{i_1,k_1}, x_{i_2,k_2}, x_{i_3,k_3}\}$ is allocated to one of $\{\alpha_i, \beta_i, \gamma_i\}$ (i.e., one of the variables $x_{i_1}, x_{i_2}, x_{i_3}$ is set "true"), and
- either all or none of $\{x_{i,1}, x_{i,2}, x_{i,3}\}$ are allocated to some agents of the set $\mathcal{C} = \{\alpha_j, \beta_j, \gamma_j | 1 \leq i, j \leq n\}$, i.e., either $\{x_{i,1}, x_{i,2}, x_{i,3}\} \subset (\bigcup_{a \in \mathcal{C}} P(a))$ or $\{x_{i,1}, x_{i,2}, x_{i,3}\} \cap (\bigcup_{a \in \mathcal{C}} P(a)) = \emptyset$ holds.

Therewith, the truth assignment ϕ which sets x_i "true" if and only if $x_i \in P(X_i)$ (i.e., $x_{i,1}, x_{i,2}, x_{i,3}$ are allocated to some agent $\{\alpha_j, \beta_j, \gamma_j\}$, $1 \leq j \leq n$), is a feasible truth assignment that sets "true" exactly one variable of each clause. Hence, $\mathcal{I}$ is a "yes"-instance of CUBIC MONOTONE 1-IN-3 SAT. $\square$

4 Conclusion

We have shown that maximizing Nash product social welfare is computationally intractable when Borda or Lexicographic scores are used, and solvable in polynomial time for Approval scores. An

interesting open question is the computational complexity of maximizing Nash product social welfare for Quasi-Indifference scores.
The NP-completeness results for Borda and Lexicographic scores imply that the problem of finding an allocation that maximizes Nash product social welfare is an NP-hard problem in these cases. A further interesting direction for future research is to investigate the existence of approximation algorithms for the problem of finding such an allocation that run in polynomial time.

References

Ahuja, R., Magnanti, T., & Orlin, J. (1993). *Network flows: theory, algorithms, and applications.* New Jersey: Prentice Hall.

Baumeister, D., Bouveret, S., Lang, J., Nguyen, T. T., Rothe, J., & Saffidine, A. (2013). Positional scoring rules for the allocation of indivisible goods. In *11th European Workshop on Multi-Agent Systems (EUMAS)* (pp. 1–14). Toulouse, France.

Brandt, F., Conitzer, V., & Endriss, U. (2013). Computational social choice. In G. Weiss (Ed.), *Multiagent Systems* (pp. 213–283). MIT Press.

Chevaleyre, Y., Dunne, P. E., Endriss, U., Lang, J., Lemaitre, M., Maudet, N., Padget, J., Phelps, S., Rodriguez-Aguilar, J. A., & Sousa, P. (2006). Issues in multiagent resource allocation. *Informatica*, 30, 3–31.

Chevaleyre, Y., Endriss, U., Estivie, S., & Maudet, N. (2008). Multiagent resource allocation in k-additive domains: preference representation and complexity. *Annals of Operations Research*, 163(1), 49–62.

Dunne, P. E., Wooldridge, M., & Laurence, M. (2005). The complexity of contract negotiation. *Artificial Intelligence*, 164(1-2), 23–46.

Golovin, D. (2005). *Max-min fair allocation of indivisible goods.* Technical report, School of Computer Science, Carnegie Mellon University.

Lipton, R. J., Markakis, E., Mossel, E., & Saberi, A. (2004). On approximately fair allocations of indivisible goods. In *Proceedings of the 5th ACM Conference on Electronic Commerce*, EC '04 (pp. 125–131). New York, NY, USA: ACM.

Moore, C. & Robson, J. (2001). Hard tiling problems with simple tiles. *Discrete & Computational Geometry*, 26(4), 573–590.

Moulin, H. (2003). *Fair division and collective welfare.* MIT Press.

Nguyen, N.-T., Nguyen, T., Roos, M., & Rothe, J. (2014). Computational complexity and approximability of social welfare optimization in multiagent resource allocation. *Autonomous Agents and Multi-Agent Systems*, 28(2), 256–289.

Ramezani, S. & Endriss, U. (2010). Nash social welfare in multiagent resource allocation. In E. David, E. Gerding, D. Sarne, & O. Shehory (Eds.), *Agent-Mediated Electronic Commerce. Designing Trading Strategies and Mechanisms for Electronic Markets*, volume 59 of *Lecture Notes in Business Information Processing* (pp. 117–131). Springer Berlin Heidelberg.

Roos, M. & Rothe, J. (2010). Complexity of social welfare optimization in multiagent resource allocation. In W. van der Hoek, G. A. Kaminka, Y. Lespérance, M. Luck, & S. Sen (Eds.), *AAMAS* (pp. 641–648).: IFAAMAS.

Andreas Darmann
Institute of Public Economics
University of Graz, Austria
Email: andreas.darmann@uni-graz.at

Joachim Schauer
Institute of Statistics and Operations Research
University of Graz, Austria
Email: joachim.schauer@uni-graz.at