MATHEMATICS OF OPERATIONS RESEARCH
Vol. 00, No. 0, Xxxxx 0000, pp. 000–000
ISSN 0364-765X | EISSN 1526-5471 | 00 | 0000 | 0001

INFORMS
DOI 10.1287/xxxx.0000.0000

Maximizing Nash Social Welfare in 2-Value Instances: Delineating Tractability*

Hannaneh Akrami
Max Planck Institute for Informatics and Universität des Saarlandes, hakrami@mpi-inf.mpg.de

Bhaskar Ray Chaudhury
University of Illinois, Urbana-Champaign, Department of Computer Science, braycha@illinois.edu

Martin Hoefer
RWTH Aachen University, Department of Computer Science, mhoefer@cs.rwth-aachen.de

Kurt Mehlhorn
Max Planck Institute for Informatics and Universität des Saarlandes, mehlhorn@mpi-inf.mpg.de

Marco Schmalhofer
Goethe University Frankfurt, Institute for Computer Science, schmalhofer@em.uni-frankfurt.de

Golnoosh Shahkarami
Max Planck Institute for Informatics and Universität des Saarlandes, gshahkar@mpi-inf.mpg.de

Giovanna Varricchio
University of Calabria, Department of Mathematics and Computer Science, giovanna.varricchio@unical.it

Quentin Vermande
DIENS, École Normale Supérieure, Paris, qvermande@phare.normalesup.fr

Ernest van Wijland
IRIF, Paris, ernest.van.wijland@irif.fr

We study the problem of allocating a set of indivisible goods among a set of agents with *2-value additive valuations*. In this setting, each good is valued either 1 or p/q, for some fixed co-prime numbers $p, q \in \mathbb{N}$ such that $1 \leq q < p$. Our goal is to find an allocation maximizing the *Nash social welfare* (NSW), i.e., the geometric mean of the valuations of the agents. In this work, we give a complete characterization of polynomial-time tractability of NSW maximization that solely depends on the values of q.

We start by providing a rather simple polynomial-time algorithm to find a maximum NSW allocation when the valuation functions are *integral*, that is, $q = 1$. We then exploit more involved techniques to get an algorithm producing a maximum NSW allocation for the *half-integral* case, that is, $q = 2$. Finally, we show it is NP-hard to compute an allocation with maximum NSW whenever $q \geq 3$.

Key words: Game Theory, Fair Division, Nash Social Welfare, 2-Value instances
MSC2000 subject classification:
OR/MS subject classification: Primary: Game Theory; secondary:
History:

1. Introduction Fair division of goods has developed into a fundamental field in economics and computer science. In a classical fair division problem, the goal is to allocate a set of goods among a set of agents in a *fair* (making every agent satisfied with her bundle) and *efficient* (achieving good overall welfare) manner. One of the most well-studied classes of valuation functions is the one of *additive* valuation functions, where the utility of a bundle is defined as the sum of the utilities of the contained goods. When agents have additive valuation functions, Nash social welfare (NSW), or equivalently the geometric mean of the valuations, is a direct indicator of the fairness and efficiency of an allocation. In particular, Caragiannis et al. [14] show that any allocation that maximizes NSW is *envy-free up to one good* (EF1), i.e., no agent envies another agent after the removal of *some* single good from the other agent's bundle, and *Pareto-optimal*, i.e., no allocation gives a single agent a better bundle without giving a worse bundle to some other agent. Lee [31] shows that maximizing NSW is APX-hard and allocations that achieve good approximations of Nash social welfare may not have similar fairness and efficiency guarantees. Despite this, in the past years, several algorithms with small constant approximation factors were obtained, Anari et al. [5], Barman et al. [9], Cole et al. [18], Cole et al. [19]. The current best factor is $e^{1/e} \approx 1.445$, Barman et al. [9]. The provided algorithm uses prices and techniques inspired by competitive equilibria, along with suitable rounding of valuations to guarantee polynomial running time.

* An extended abstract of this paper has been published in the proceedings of the AAAI 2022 conference, Akrami et al. [2], with the title "Maximizing Nash Social Welfare in 2-Value Instances". The adjusted title emphasizes our complete characterization of the tractable cases in terms of the denominator of the ratio of integer item values.

While computing an allocation with maximum NSW is generally hard, it becomes computationally tractable when the agents have *binary additive valuations*, i.e., when for each agent i and each good g, we have $v_i(g) \in \{0,1\}$, Barman et al. [10]. Although this class of valuation functions may seem restrictive in its expressiveness, several real-world scenarios involve binary preferences and, in fact, there is substantial research on fair division under binary valuations, Aleksandrov et al. [3], Barman et al. [10], Bouveret et al. [11], Darmann et al. [21], Freeman et al. [24], Halpern et al. [30], Suksompong et al. [36]. In this setting, Barman et al. [10] gave a polynomial-time algorithm to find an allocation with maximum NSW. Furthermore, Halpern et al. [30] show that determining a fair allocation via NSW maximization is also *strategyproof*, i.e., agents do not benefit by misreporting their real preferences. Such results have been extended to the case of asymmetric agents by Suksompong et al. [36], i.e, agents with different entitlements. Indeed, they show that, for binary additive valuations and asymmetric agents, a maximum NSW allocation satisfies strategyproofness, together with other interesting properties, and can be computed in polynomial time.

A generalization of binary valuation functions are *2-value* functions, where for each agent i and each good g, we have $v_i(g) \in \{a,b\}$ for some $a,b \in \mathbb{Q}$. W.l.o.g. we may assume $a=1$ and $b>a$. The case $a=b$ is trivial as every agent gives the same value to all the goods. Binary valuations are the special case $a=0$ and $b=1$. Amanatidis et al. [4] show that for 2-value functions, an allocation with maximum NSW is *envy-free up to any good* (EFX), where no agent envies another agent following the removal of *any* single good from the other agent's bundle. The authors also provide a polynomial-time algorithm providing an EFX allocation for 2-value instances. However, the computed allocation is neither guaranteed to maximize NSW nor to be Pareto optimal. Garg and Murhekar [29] show how to obtain an EFX and PO allocation efficiently and also provide a 1.061-approximation algorithm for the maximum NSW in 2-value instances; this was improved to 1.0345 in the conference version of this paper [2]. Both Amanatidis et al. [4] and Garg and Murhekar [29] left the problem of (exactly) maximizing Nash social welfare for 2-value instances open. For 3-value instances maximizing Nash social welfare is NP-complete [4].

1.1. Our Contribution In this paper, we solve the problem of maximizing Nash social welfare for 2-value instances. Surprisingly, the tractability of this problem changes according to the ratio between the two values a and b. Since scaling an agent's valuation by a uniform factor for all goods does not affect the optimality properties of allocations, let us assume w.l.o.g. that $a=1$ and $b={}^p/_q$, for some coprime numbers $p,q \in \mathbb{N}$ such that $1 \le q < p$.

We first show two positive results. If q is either 1 or 2, there is a polynomial-time algorithm computing a maximum NSW allocation.

THEOREM 1. *There exists a polynomial-time algorithm computing a maximum NSW allocation for integral instances, i.e., when $q = 1$ and p is an integer greater than one.*

THEOREM 2. *There exists a polynomial-time algorithm computing a maximum NSW allocation for half-integral instances, i.e., when $q = 2$ and p is an odd integer greater than two.*

The proof and the algorithm for the half-integer result is considerably more complex than the ones for the integer case. We complete the characterization in terms of computational complexity by showing NP-hardness in the remaining cases, i.e., when $q \geq 3$.

THEOREM 3. *It is NP-hard to compute an allocation with optimal NSW for 2-value instances, for any constant coprime integers $p > q \geq 3$.*

The analysis of our algorithms requires a number of novel technical contributions. In the next section, we highlight the main ideas and techniques.

1.2. Our Techniques We interpret the problem of maximizing NSW as a graph problem: We have a weighted complete bipartite graph with the set of agents and the set of goods being the independent sets. The edge weights represent the value of a good for an agent. They are either 1 (in which case we call the edge a *light edge*) or p/q (*heavy edge*). We say that a good is heavy if it has at least one incident heavy edge and light otherwise. A (partial) allocation is a multi-matching in which every good has a degree of at most one. We call an allocation complete if all goods have degree one. This representation of allocations allows us to use the idea of alternating paths, similar to the algorithm proposed by Barman et al. [10] for instances with binary valuations.

However, there are also crucial structural differences between 2-value instances and binary instances. For binary instances, Halpern et al. [30] show that an allocation maximizes NSW if and only if it is *lexmax*, i.e., the utility profile of the allocation is lexicographically maximum.[1] This is not true for 2-value instances. Consider the following example: There are two agents, 1 and 2, five goods g_1, g_2, g_3, g_4, and g_5, and heavy goods have value 5. Agent 1 values g_1 and g_2 as heavy, and the remaining goods as light. For agent 2, all goods are light. In an optimal allocation agent 1 gets $\{g_1, g_2\}$ and agent 2 gets $\{g_3, g_4, g_5\}$. However, this allocation is not lexmax, as it is lexicographically dominated by the allocation where agent 1 gets $\{g_1\}$ and agent 2 gets $\{g_2, g_3, g_4, g_5\}$. This requires finding some *other tractable characterization* of the allocations with maximum NSW, in particular, a characterization of the allocation of heavy goods in such an allocation.

Another crucial observation is that the structure of the heavy goods in an optimal allocation may depend on the ratio between heavy and light goods' values. To show how the allocation of

[1] For an allocation $A = (A_1, \ldots, A_n)$, the utility profile is the vector $(v_1(A_1), \ldots, v_n(A_n))$ sorted into non-decreasing order. A profile is lexmax if there is no other profile that it lexicographically larger.

heavy goods changes in optimal allocations for different p/q values, let us modify the valuations in the previous example instance with two agents and five goods.

EXAMPLE 1. Suppose the two agents have identical valuations. Goods g_1 and g_2 are heavy, the remaining goods are light. Heavy goods have value p/q and light goods have value 1. It is easy to see that any optimal solution follows one of two patterns: (1) assign both heavy goods to one agent, all light goods to the other agent; (2) assign one heavy good to each agent, and in addition, one agent gets two light goods, the other gets one light good. Whether optimal allocations follow the first and/or the second pattern depends on the ratio between p and q. In particular, it is easy to see that all optimal allocations follow the first pattern if and only if $p/q < 2$; all follow the second one if and only if $p/q > 2$. There are optimal allocations for both patterns when $p/q = 2$. Hence, depending on the ratio p/q, the distribution of heavy and light goods in optimal allocations may change. In particular, the first pattern yields an unbalanced allocation of heavy goods – one agent receives both heavy goods while the other none. In contrast, in the second pattern the heavy goods are balanced. ■

It turns out that understanding how heavy goods are distributed in optimal allocations is the key challenge in computing a maximum NSW allocation. We characterize the allocation of heavy goods and use these insights to design efficient algorithms.

Characterizing the allocation of heavy goods. First, we consider instances with $q = 1$ (which we call *integral* instances). We give a concise characterization of the distribution of heavy goods in a maximum NSW allocation. We refer to the *heavy-part* A^H of an allocation A as the set of all heavy edges in the allocation, and we call an allocation *heavy-only* if the allocation contains only heavy edges, i.e., if $A^H = A$. One of our main structural results (shown in Lemma 2) is that there exists a maximum NSW allocation OPT, such that the heavy-part of OPT is lexicographically maximum (*lexmax*) among all heavy-only allocations of the same cardinality. Therefore, if we know the number of heavy-edges in OPT, then the utility profile of the heavy-part of OPT is unique (as it is lexmax).

For instances with $q = 2$ (called *half-integral* instances), the lexmax property is not necessarily satisfied. The lexmax property may be interpreted as "the distribution of heavy goods is balanced as much as possible". Here, however, the heavy-part of the allocation might have to be unbalanced in order to maximize the NSW (see Example 1). For this reason, we present a local search algorithm using a number of *improvement rules* that allow us to redistribute heavy and light goods. Such improvement rules tend to unbalance the heavy-part but also increase the NSW of the allocation. For the efficient realization of the improvement rules we exploit a connection to generalized bipartite matching. Consider a bipartite graph and let A be one of the sides of the graph. In a generalized matching problem the goal is to find a subset of the edges such that the degree of any node $a \in A$ is in a prescribed set $D(a)$. If the gaps (distance between consecutive values) in the D-sets are at most two, generalized matching problems are polynomially solvable [33, 20].

Finding the right allocation of heavy goods. The crucial technical barrier lies in the fact that we do not know the number of heavy edges in OPT. We briefly elaborate on how we overcome this barrier. The key is to solve the computation of a maximum NSW allocation for a fixed number of goods assigned as heavy. For both $q = 1$ and $q = 2$, we show this problem is polynomial-time solvable. Therefore, our algorithms proceed in sequential phases. They start from an allocation where every heavy good is allocated as heavy, that is, we start from an allocation where the number of allocated heavy goods is maximized and produce an optimal allocation. In each of the subsequent steps, we reduce the number of goods allocated as heavy by one by first converting an arbitrary heavy good in a heaviest bundle to light and then re-optimizing.

Hardness results. For instances with $q \geq 3$, it is NP-hard to compute a maximum NSW allocation. Our proof is based on a reduction from Exact q-Dimensional Matching. Our proof formalizes the intuition that *determining the structure of heavy goods in an optimal allocation* is the main challenge in maximizing the NSW. In particular, we show that it is NP-hard to determine the distribution of heavy goods in an optimal allocation.

1.3. Further Related Work Our algorithm for integral instances and the NP-hardness results in the current paper have been presented as part of an extended abstract in the proceedings of the AAAI'22 conference, Akrami et al. [2], where we also discussed APX-hardness for $q \geq 4$ and showed that our algorithm for integral instances has an approximation factor of at most $\frac{24}{29} \exp\left(\frac{110}{493}\right) < 1.0345$ for general 2-value instances. For the present paper, we decided to focus on the computational complexity of optimal solutions and to omit the consideration of approximation. In particular, we provide a new efficient algorithm for half-integral instances, which was left as an open problem in Akrami et al. [2].

Beyond additive valuations, the design of approximation algorithms for *submodular valuations* received considerable attention. While small constant approximation factors have been obtained for special cases by Anari et al. [6], Garg et al. [25], and Chaudhury et al. [15]), such as a factor $e^{1/e}$ for capped additive-separable concave valuations, only rather high constants for Rado valuations, Garg et al. [27], and also general non-negative, non-decreasing submodular valuations, Li et al. [32], have been obtained. The currently best approximation ratio for submodular valuations is $4 + \epsilon$ by Garg et al. [26].

Interestingly, for *binary* submodular valuations where the marginal value of every agent for every good is either 0 or 1, an allocation maximizing the NSW can be computed in polynomial time, Babaioff et al. [7]. In particular, in this case, one can in polynomial time find an allocation that is Lorenz dominating, simultaneously minimizes the lexicographic vector of valuations, and maximizes both utilitarian social welfare, i.e., the sum of the agents' utilities, and NSW. Moreover, this allocation is also strategyproof.

More generally, there are approximation algorithms for maximizing NSW with subadditive valuations, Barman et al. [8], Chaudhury et al. [16], Dobzinski et al. [22], and even asymmetric agents, Garg et al. [28] and Brown et al. [12].

There is also literature on guaranteeing high NSW with other fairness notions. For instance, relaxations of EFX can be guaranteed with high NSW, Caragiannis et al. [13] and Chaudhury et al. [16]. Moreover, approximations of *groupwise maximin share* (GMMS), Chaudhury et al. [17], and *maximin share* (MMS), Caragiannis et al. [14] and Chaudhury et al. [17], are achieved with high NSW.

1.4. Organization The rest of this paper is structured as follows. We start by providing preliminary definitions and notations in Section 2. In Section 3, we discuss the polynomial-time algorithm computing a maximum NSW allocation for integral instances. In Section 4, we elaborate more involved techniques to fulfill the same goal in the case of half-integral instances. Finally, in Section 5, we show that for other classes of 2-value instances the problem of maximizing the NSW is NP-hard.

2. Preliminaries A fair division instance $\mathcal{I}$ is given by a triple (N, M, v), where N is a set of $n \geq 1$ agents and M is a set of $m \geq n$ indivisible goods. Every agent $i \in N$ has an additive valuation function $v_i : 2^M \to \mathbb{R}_{\geq 0}$, with $v_i(X) = \sum_{g \in X} v_i(\{g\})$, for every $X \subseteq M$. For the sake of simplicity, we use $v_i(g)$ instead of $v_i(\{g\})$. Agents' valuations are stored in the *valuation vector* $v = (v_1, \ldots, v_n)$.

In this paper, we study *2-value* additive valuations, in which, for each $g \in M$, $v_i(g) \in \{1, {}^p/_q\}$ for fixed $p, q \in \mathbb{N}$. To avoid trivialities, we assume $0 < q < p$, and p, q to be coprime numbers. Note that for $p = 0$ one could recover the binary case studied in Babaioff et al. [7] and Barman et al. [10].

An *allocation* $A = (A_1, \ldots, A_n)$ is a partition of M among the agents, where $A_i \cap A_j = \emptyset$, for each $i \neq j$, and $\bigcup_{i \in N} A_i = M$. We evaluate an allocation using the Nash social welfare

$$\mathrm{NSW}(A) = \left(\prod_{i \in N} v_i(A_i) \right)^{1/n}.$$

We denote by $\mathrm{NSW}(A, v)$ the NSW of the allocation A under the utility vector v. In case v is clear from the context, we simply write $\mathrm{NSW}(A)$.

(a) Graph representation. (b) An allocation.

FIGURE 1. The depicted graph corresponds to an instance with two agents and three goods. Thick black edges and thin gray edges correspond to heavy and light edges, respectively.

2.1. Utility Graphs and Utility Profiles In our paper, we exploit the connection between the concepts of allocation in fair division and of one-to-many matchings in bipartite graphs. For this reason, we find it more convenient to define an allocation in the context of a bipartite graph. Consider the complete bipartite graph $G = (N \cup M, E)$, where we have agents on one side and goods on the other side. We call the edge between agent i and good g *heavy*, if $v_i(g) = p/q$ and *light* otherwise. We use E^H and E^L to denote the set of heavy and light edges, respectively. Moreover, good g is *heavy* for agent i if $v_i(g) = p/q$ and *light* otherwise. In Figure 1a we provide an instance with two agents and three goods.

An *allocation* is a subset $A \subseteq E$ such that for each $g \in M$ there is at most one edge in A incident to g. Note that according to this definition allocations may be partial. If there is an edge $(i, g) \in A$, we say that g is *assigned* to i in A, or i *owns* g in A, or A *assigns* g to i. Otherwise, g is *unassigned*. An allocation is *complete* if all goods are assigned. For an agent i, we use A_i to denote the set of goods assigned to i in A. We call A_i the *bundle* of i in A. Then $v_i(A_i)$ is the utility of i's bundle for i. Figure 1b shows an allocation for the instance depicted in Figure 1a.

The *utility vector* of an allocation A is given by $(v_1(A_1), \ldots, v_n(A_n))$, and its *utility profile* is obtained by rearranging its components in non-decreasing order. A utility profile $(a_1, \ldots, a_n)$ is *lexicographically larger* than a utility profile $(b_1, \ldots, b_n)$ (denoted by $(a_1, \ldots, a_n) \succ_{lex} (b_1, \ldots, b_n)$) if the profiles are different and $a_i > b_i$ for the smallest i with $a_i \neq b_i$. An allocation A with utility profile $(a_1, \ldots, a_n)$ is *lexmax* in a family $\mathcal{A}$ of allocations if there is no allocation $B \in \mathcal{A}$ with utility profile $(b_1, \ldots, b_n)$ such that $(b_1, \ldots, b_n) \succ_{lex} (a_1, \ldots, a_n)$.

For an allocation A, its *heavy part* A^H is the restriction of A to the heavy edges, i.e., $A^H = A \cap E^H$. An allocation A is *heavy-only* if $A = A^H$. For an agent i, A_i^H is the set of heavy edges incident to agent i under allocation A. We refer to $|A_i^H|$ as the *heavy degree* of i in A and denote it by $\deg_H(i, A)$ or $\deg_H(i)$.

2.2. Alternating Paths We reformulated the fair division setting so that allocations correspond to multi-matchings. This is motivated by the fact that, in our algorithms, we improve the NSW of an allocation using the notion of *alternating paths*.

(a) Allocation A and the heavy alternating path P. (b) $A \oplus P$.

FIGURE 2. In both figures the path $P = (g_1, 1, g_2, 2, g_3)$ is depicted. In Figure 2, solid edges represent the allocation.

An *alternating path* with respect to an allocation A is any path whose edges alternate between A and $E \setminus A$. Alternating paths having only heavy edges will be of particular interest. A *heavy alternating path* is an alternating path whose edges belong to E^H. In Figure 2a, we give an example of a heavy alternating path for the instance depicted in Figure 1a.

An *alternating path* with respect to two allocations A and B is any path whose edges alternate between $A \setminus B$ and $B \setminus A$, i.e., between edges only in A and edges only in B.

An *alternating path decomposition* is defined with respect to two heavy-only allocations A and B. The graph $A \oplus B$ is defined on the same set of vertices as in A and B. Moreover, an edge e appears in $A \oplus B$, if and only if e is in exactly one of A or B. We decompose $A \oplus B$ into edge-disjoint paths; this decomposition is not unique. Note that in $A \oplus B$, goods have degree zero, one, or two. For a good of degree two, the two incident edges belong to the same path. For an agent i, let a_i, respectively b_i, be the number of A-edges, respectively B-edges, incident to i in $A \oplus B$. Then we have $\min(a_i, b_i)$ alternating paths passing through i, $\max(0, a_i - b_i)$ alternating paths starting in i with an edge in A, and $\max(0, b_i - a_i)$ alternating paths starting in i with an edge in B. The paths in the decomposition are maximal in the sense that no path can be extended without breaking another one.

Let P be a heavy alternating path with respect to A of even length. Assume P connects two agents i and j with the edge of P incident to i in $E^H \setminus A$ and the edge incident to j in A^H. Then $A \oplus P$ contains the same number of heavy edges as A, i.e., $|A^H| = |(A \oplus P)^H| = |A^H \oplus P|$. Moreover, the heavy degree of i increases by one, the heavy degree of j decreases by one, and all other heavy degrees are unchanged.

EXAMPLE 2. Consider the example shown in Figure 1a and allocation A with

$$A^H = \{(1, g_1), (2, g_2)\}$$

shown in Figure 1b. Let B be another allocation for which

$$B^H = \{(1, g_1), (1, g_2)\}.$$

Figure 3 shows $A^H \oplus B^H$. Black edges are only in A^H and dashed edges are only in B^H. The path decomposition of $A \oplus B$ consists of the unique path $P = (1, g_2, 2)$ in $A \oplus B$. ■

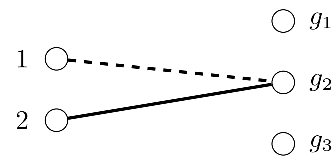

FIGURE 3. $A^H \oplus B^H$

We will often compare distinct allocations and use a notion of distance between them. The *distance* between two allocations is the number of edges that only exist in one of the allocations; formally, the distance between two allocations A and B is $|A \oplus B|$.

It will turn out that in half-integral instances dealing with alternating paths is not sufficient to compute optimal allocations. For this reason, in Section 4 below, we will also rely on a more involved path structure to deal with this problem; namely, *improving walks*.

In the rest of the paper, we denote by OPT an allocation maximizing the NSW and by OPT^H its heavy part. Furthermore, we specify the bundle of the agent i in OPT by OPT_i and denote its heavy part by OPT_i^H.

2.3. Math Preliminaries The following Lemma is useful for showing that certain re-allocations increase the NSW.

LEMMA 1. *Let a, b, c, d, d_1, and d_2 be non-negative reals.*

a). *If $a \ge b$ and $d \in [0, a-b]$ then $ab \le (a-d)(b+d)$ with equality if and only if $d = 0$ or $d = a-b$.*

b). *If $a \ge b \ge c$, $b \ge c + d_2$, and $a \ge c + d_1 + d_2$ then $abc \le (a-d_1)(b-d_2)(c+d_1+d_2)$ with equality if and only if $c = 0$ and $d_2 = b$ or $d_2 \in \{0, b-c\}$ and $d_1 \in \{0, a-c-d_2\}$.*

Proof. For a) we have $a \ge b + d \ge b$ and $d \ge 0$ and hence

$$(a-d)(b+d) - ab = (a-b-d)d \ge 0$$

with equality if and only if $d = 0$ or $d = a-b$.

Part b) is obvious if $a = 0$. Note that $a = 0$ implies $b = c = d_1 = d_2 = 0$. It is also obvious, if $c = 0$ and $d_2 = b$. Then LHS and RHS are zero. So assume $a > 0$ and either $c > 0$ or $d_2 < b$. In either case $b - d_2 > 0$. We apply part a) twice and obtain

$$\begin{aligned} abc &\le a(b-d_2)(c+d_2) && \text{with equality iff } d_2 = 0 \text{ or } d_2 = b-c \text{ since } a > 0. \\ &\le (a-d_1)(b-d_2)(c+d_1+d_2) && \text{with equality iff } d_1 = 0 \text{ or } d_1 = a-c-d_2 \text{ since } b-d_2 > 0. \end{aligned}$$

Q.E.D.

3. Integral Instances In this section, we consider *integral* instances, i.e., $q = 1$ and p an integer greater than one. We also term the valuation functions integral. Our main result is a polynomial-time algorithm to find a maximum NSW allocation.

3.1. Properties of an Optimal Allocation We first study the properties of optimal allocations. The main insight is stated in Lemma 2. Roughly speaking, it states that there exists an optimal allocation OPT, in which heavy goods are assigned as evenly as possible. More formally, the utility profile of OPT^H is lexmax among all heavy-only allocations with the same cardinality. Later, we use this property to prove that the utility profile of A^H at the end of Algorithm 1 is equal to the utility profile of OPT^H, if OPT is chosen cleverly among the set of all optimal allocations. After this, it will not be difficult to prove that the utility profiles of A and OPT match.

For an allocation A, $\min(A) = \min_i v_i(A_i)$ denotes the minimum utility of any of its bundles.

Claim 1. *Let OPT be an optimal allocation and let j be an agent. If $v_j(OPT_j) \geq \min(OPT) + 2$ then all goods in OPT_j are heavy for j.*

Proof. Assume otherwise, and take a good that is light for j and reallocate it to an agent i for which $v_i(\mathrm{OPT}_i) = \min(\mathrm{OPT})$. This will improve the NSW by Lemma 1a). Q.E.D.

Corollary 1. *Let OPT be any optimal allocation. Only bundles of utility $\min(OPT)$ and $\min(OPT) + 1$ can contain light goods. Bundles with higher values only contain goods that are heavy for their owner.*

Claim 2. *If there is a heavy alternating path with respect to OPT starting with an OPT-edge from an agent i to an agent j then $v_i(OPT_i) \leq v_j(OPT_j) + p$.*

Proof. Assume such an alternating path exists and call it P. In $\mathrm{OPT} \oplus P$, i is incident to one fewer heavy edge and j is incident to one more heavy edge, and hence the NSW changes by the factor $(v_i(\mathrm{OPT}_i) - p)(v_j(\mathrm{OPT}_j) + p)/v_i(OPT_i)v_j(OPT_j)$. This factor must be no larger than one. Thus $v_i(OPT_i) \leq v_j(OPT_j) + p$. Q.E.D.

Claim 3. *If a good g is allocated as a light good to an agent i but could be allocated as a heavy good to an agent j who is allocated good g' which is light for j, then the allocation is not optimal.*

Proof. Swapping the goods g and g' among agent i and agent j increases the value of agent j by $p - 1$ and does not decrease the value of agent i. Q.E.D.

The rest of this section is dedicated to proving the following lemma.

Lemma 2. *Among all allocations with maximum NSW, there exists an allocation A such that the utility profile of A^H is lexmax among all heavy-only allocations of the same cardinality.*

Proof. We choose A and heavy-only C^H as follows: (1) A is an optimal allocation, (2) C^H is lexmax among all allocations of $|A^H|$ heavy goods, and (3) the distance between A^H and C^H is minimum among all allocations satisfying (1) and (2).

We will show that A^H and C^H agree. Assume otherwise and let us consider $A^H \oplus C^H$. We label an edge with either A or C indicating whether it belongs to A^H or C^H. Note that in this graph, goods have degree zero, one, or two. $A^H \oplus C^H$ decomposes into edge-disjoint maximal alternating paths and cycles. We first show that there are no heavy alternating cycles.

Observation 1. There are no heavy alternating cycles.

Proof. Assume first that there is an alternating cycle, say D. Then $C^H \oplus D$ has the same utility profile as C^H and is closer to A^H, a contradiction. Q.E.D.

Hence, we have only alternating paths. We next make more subtle observations about the edge-disjoint alternating paths in $A^H \oplus C^H$. First, we show that we cannot have an even-length alternating path with both endpoints as goods.

Observation 2. There are no maximal even-length heavy alternating paths with both endpoints as goods.

Proof. Assume otherwise and let P be such a path. Then $C^H \oplus P$ has the same utility profile as C^H and is closer to A^H, a contradiction. Q.E.D.

So at least one endpoint of each maximal alternating path is an agent. If there is an even length maximal alternating path, both endpoints are agents. Let P be such a path, let i and j be its endpoints, and assume w.l.o.g. that P starts in i with an edge in A^H and ends in j with an edge in C^H. Then $|A_i^H| > |C_i^H|$ and $|C_j^H| > |A_j^H|$. Set Q to the empty path.

If all maximal alternating paths have odd length, exactly one endpoint of each path is an agent. Let i and j be agents with $|A_i^H| > |C_i^H|$ and $|C_j^H| > |A_j^H|$ respectively, and let P and Q be maximal alternating path starting in i and j respectively. The other endpoints of P and Q are goods.

We next show $|A_j^H| \le |A_i^H| - 2$.

Observation 3. $|A_j^H| < |A_i^H|$.

Proof. Assume otherwise, i.e., $|A_j^H| \ge |A_i^H|$. Then $|C_j^H| > |C_i^H| + 1$ and hence $C^H \oplus P \oplus Q$ is lexicographically larger than C^H, a contradiction. Q.E.D.

Observation 4. $|A_j^H| < |A_i^H| - 1$.

Proof. If $|A_j^H| = |A_i^H| - 1$, then $A^H \oplus P \oplus Q$ and A^H have the same utility profile with respect to heavy goods. Also $A^H \oplus P \oplus Q$ is closer to C^H than A^H. Finally, we swap the goods that are light for i in A_i with the goods that are light for j in A_j. The value of the resulting bundles for i and j are at least $v_j(A_j)$ and $v_i(A_i)$, respectively. If either inequality is strict, A was not optimal, a contradiction. So both inequalities are equalities, and thus the resulting allocation is again optimal, and with respect to heavy edges, it has the same utility profile as before and is closer to C^H, a contradiction. Q.E.D.

OBSERVATION 5. The paths P and Q do not exist.

Proof. By Observation 4, we have $|A_j^H| \le |A_i^H| - 2$. By Claim 2, $v_i(A_i) \le v_j(A_j) + p$, and hence A_j contains p goods light for j. Augmenting P and Q to A and moving these light goods to A_i yields an allocation that has the same NSW as A and is closer to C^H, contradicting the choices of A and C^H. Q.E.D.

We can now complete the proof of Lemma 2. Since P and Q do not exist, $A^H = C^H$. Q.E.D.

COROLLARY 2. *Among all allocations maximizing the NSW, let OPT be such that OPT^H is lexmax among the heavy parts of allocations maximizing the NSW. Then OPT^H is lexmax among all heavy-only allocations B^H with $|B^H| = |OPT^H|$.*

LEMMA 3. *Let OPT be an optimal allocation. Let L be the set of goods that are allocated as light goods in OPT. Then the following allocation is also optimal. Start with OPT^H (or any allocation of the items in OPT^H with the same utility profile) and then allocate the goods in L greedily, i.e., allocate the goods one by one and for each good $g \in L$ choose an arbitrary agent i that currently owns a bundle of minimum value and assign g to her.*

Proof. Assume otherwise. Then there is a last allocation in the sequence of partial allocations that can be extended to an optimal allocation. Let X be this partial allocation. Greedy allocates the next light good g to agent i, but after the allocation of g to i the new allocation cannot be extended to an optimal allocation. So $v_i(\mathrm{OPT}_i) \le v_i(X_i)$ for all optimal allocations OPT. Also $v_i(X_i) \le v_j(X_j)$ by the choice of i and $v_j(X_j) \le v_j(\mathrm{OPT}_j^*)$ for some optimal allocation OPT^* since X can be extended to an optimal allocation. Since g is not allocated to i in OPT^*, there must be an agent k such that $v_k(X_k) + 1 \le v_k(\mathrm{OPT}_k^*)$ and OPT_k^* contains a light good. Now modify OPT^* by moving a light good from OPT_k^* to OPT_i^*. The move does not decrease the Nash social welfare, since $v_i(\mathrm{OPT}_i^*) \le v_i(X_i) \le v_k(X_k) \le v_k(\mathrm{OPT}_k^*) - 1$, and creates an optimal allocation extending X. Q.E.D.

Before we move on to explain the algorithm, we remind the reader of Example 1 showing that this approach fails when p/q is not an integer. In particular, the example shows that Lemma 2 is not true in the half-integral case, where $p/q = 3/2$.

3.2. Algorithm In this subsection, we describe and analyze Algorithm 1. It operates in three phases.

The first phase finds a heavy-only allocation that maximizes the NSW. This phase is equivalent to maximizing the NSW in a binary instance. Barman et al. [10] proved that this is possible in polynomial time. Notice that, after this phase, light goods remain unallocated.

Algorithm 1: TwoValueMaxNSW

Input : $N, M, v = (v_1, ..., v_n)$

Output: allocation A

`/* Phase 1: Heavy-Only Allocation */`

let $v'_i : 2^M \to \mathbb{N}$ be an additive function with $v'_i(g) = 1$ if $v_i(g) = p$ and $v'_i(g) = 0$ otherwise for all $i \in N$, $g \in M$

$v' \leftarrow (v'_1, \ldots, v'_n)$

$A = \text{BinaryMaxNSW}(N, M, v')$

`/* Phase 2: Allocating Light Goods */`

number the agents so that $v_1(A_1) \leq v_2(A_2) \leq ... \leq v_n(A_n)$

while *there is an unallocated good* g **do**

  `/*` $v_1(A_1) \leq v_2(A_2) \leq ... \leq v_n(A_n)$ `*/`

  let k be the maximum index s.t. $v_k(A_k) = v_1(A_1)$

  $A \leftarrow A \cup \{(k, g)\}$

`/* Phase 3: Increasing NSW */`

while $v_n(A_n) > p \cdot v_1(A_1) + p$ **do**

  let k be the maximum index s.t. $v_k(A_k) = v_1(A_1)$

  let t be the minimum index s.t. $v_t(A_t) = v_n(A_n)$

  let g be a good such that $(t, g) \in A$

  $A \leftarrow A \setminus \{(t, g)\}$

  $A \leftarrow A \cup \{(k, g)\}$

return A

In the second phase, we greedily allocate the remaining goods (one by one) to an agent with minimum utility. Note that all these goods are light for all agents. Otherwise, the output of the first phase does not maximize NSW among all heavy-only allocations. As such, we term this phase "allocating light goods".

In the third phase, we try to improve the NSW by re-allocating heavy goods to agents considering them light. More precisely, we take a heavy good from the bundle of an agent with maximum utility and allocate it to an agent with minimum utility as long as the NSW increases. In Lemma 6 below we show that the reallocated goods are light for their new owners. This means that as long as there is progress, we turn some heavy good into a light good.

3.2.1. Heavy-Only Allocations In the first step, we compute a heavy-only allocation maximizing the NSW. For completeness, let us recapitulate Algorithm 2 from Barman et al. [10].

In order to compute a heavy-only allocation maximizing NSW, we start with a heavy-only allocation A of maximum cardinality, i.e., in A any heavy good is assigned to an agent for which it is heavy. We then improve the NSW of A by augmentation of some heavy alternating paths. We search for a heavy even-length alternating path P connecting an agent i to an agent j, starting with an edge outside A and ending with an edge in A, and with the heavy degree of j at least two larger than the heavy degree of i in A. Given any such path P, we augment P to A, i.e., we update A to $A \oplus P$. When no such path can be found, the algorithm stops and returns allocation A.

Algorithm 2: BinaryMaxNSW

Input : $N, M, v = (v_1, \dots, v_n)$

Output: allocation A

let G be the corresponding graph, i.e, agent i is connected to good g iff g is heavy for i, and let A an arbitrary assignment of the items to the agents

while *there is an alternating path* $a_0, g_1, \dots, g_k, a_k$ *such that* $|A_0| \le |A_k| - 2$ **do**

  for $\ell \leftarrow k$ *to* 1 **do**

    $A \leftarrow A \setminus (a_\ell, g_\ell)$

    $A \leftarrow A \cup (a_{\ell-1}, g_\ell)$

return A

Barman et al. [10] proved that Algorithm 2 returns an allocation with maximum NSW for binary instances. Furthermore, Halpern et al. [30] proved that in binary instances the set of lexmax allocations is identical to the set of allocations with maximum NSW.

Lemma 4. *Algorithm 2 (and hence the first phase of Algorithm 1) computes an allocation A^H with maximum NSW. Furthermore, the optimal allocations of the heavy items are exactly the lexmax allocations.*

Before proceeding to the next phase, we briefly explain how lexmax heavy-only (partial) allocations of different cardinalities can be found. A heavy-only allocation maximizing the NSW is a heavy-only allocation of maximum cardinality. In order to compute heavy-only allocations of smaller cardinality, we repeatedly remove an edge from A. We take any bundle A_i with $v_i(A_i) = \max_j v_j(A_j)$ and remove an edge of A incident to i. In this way, we will obtain optimal allocations for every cardinality.

Lemma 5. *Let $C^H \subseteq E^H$ be lexmax among all allocations $D^H \subseteq E^H$ with $|D^H| = |C^H|$. Let $(c_1, c_2, \dots, c_n)$ be the utility profile of C^H and let t be such that $c_{t-1} < c_t = \dots = c_n$. Then an allocation $\hat{C}^H$ with utility profile $(\hat{c}_1, \dots, \hat{c}_n) = (c_1, \dots, c_{t-1}, c_t - p, c_{t+1}, \dots, c_n)$, is lexmax among all heavy-only allocations with the same cardinality.*

Proof. Let $\hat{D}^H$ be lexmax among all allocations with $|\hat{C}^H|$ many heavy allocated goods and let $(\hat{d}_1, \dots, \hat{d}_n)$ be the utility profile of $\hat{D}^H$. Note that since $|\hat{D}^H| = |\hat{C}^H| < |C^H|$, there exists a good g that is unallocated under $\hat{D}^H$ and is of value p for some agent a.

We need to show $\hat{D}^H = \hat{C}^H$. Assume otherwise. Consider the smallest i such that $\hat{d}_i \neq \hat{c}_i$. Then $\hat{d}_i > \hat{c}_i$ since $\hat{D}^H$ is lexmax. If $i < t$, allocating g to agent a results in an allocation D^H with $|D^H| = |C^H|$ which is lexicographically larger than C^H. This contradicts the choice of C^H.

Therefore $\hat{d}_i = \hat{c}_i$ for all $i < t$ and hence $\Sigma_{i=t}^n \hat{d}_i = \Sigma_{i=t}^n \hat{c}_i = (n-t+1)c_n - p$. Since $(c_n - p, c_n, \dots, c_n)$ is lexmax among all $(n-t+1)$ tuples with sum $(n-t+1)c_n - p$, $(\hat{d}_t, \dots, \hat{d}_n) \preceq_{lex} (c_n - p, \dots, c_n) = (\hat{c}_t, \dots, \hat{c}_n)$. Hence $\hat{C}^H$ is lexmax among all allocations $\hat{D}^H$ with $|\hat{D}^H| = |\hat{C}^H|$. Q.E.D.

Corollary 3. *Let $(p \cdot a_1, ..., p \cdot a_n)$ and $(p \cdot b_1, ..., p \cdot b_n)$ be the utility profiles of two heavy-only allocations A and B. Note that $\Sigma_{i=1}^n a_i = |A^H|$ and $\Sigma_{i=1}^n b_i = |B^H|$. Assume $|A^H| \leq |B^H|$. If the utility profile of A is lexmax among all the utility profiles of heavy-only allocations C with $|C| = |A|$ and the same holds for B, then for all $1 \leq i \leq n$, $a_i \leq b_i$.*

Proof. Keep removing goods from the bundle with maximum utility and minimum index in B until we reach an allocation $\hat{B}$ with $|\hat{B}^H| = |A^H|$. By Lemma 5, $(p \cdot \hat{b}_1, ..., p \cdot \hat{b}_n) \succeq_{lex} (p \cdot a_1, ..., p \cdot a_n)$ and therefore $(\hat{b}_1, ..., \hat{b}_n) = (a_1, ..., a_n)$. The fact that $\hat{b}_i \leq b_i$ for all $i \in [n]$, completes the proof. Q.E.D.

3.3. Correctness Phase 1 already gives us an optimal allocation X of the heavy-only goods, which is also lexmax on the allocation of the heavy-only goods. In phase 2, we allocate the light goods as "evenly" as possible. The only reason why our solution may not be optimal is that the number of heavy goods in an optimal allocation Y may be less than that in X. However, if this is the case, then we can move from X to Y by making small local improvements in NSW by moving heavy goods from one bundle to the other.

Let A be the allocation computed by Algorithm 1. First, we prove that there is an allocation OPT with maximum NSW such that the utility profile of OPT^H and A^H are the same. Then we prove that in allocation OPT, the remaining goods are allocated the same way as in A. We start by establishing some invariants of Algorithm 1.

Lemma 6. *Fix a numbering of the agents at the beginning of phase 3 such that $v_1(A_1) \leq v_2(A_2) \leq \ldots \leq v_{n-1}(A_{n-1}) \leq v_n(A_n)$. During phase 3, the following holds:*

a. *The ordering $v_1(A_1) \le v_2(A_2) \le \ldots \le v_{n-1}(A_{n-1}) \le v_n(A_n)$ is maintained.*

b. *Let i be any agent. If A_i contains a good that is light for i, then $v_i(A_i) \le v_1(A_1) + 1$.*

c. *A^H is lexmax among all heavy-only allocations of the same cardinality.*

d. *Whenever a good is moved in phase 3, say from bundle A_t to bundle A_k, all goods in A_t are heavy for t and light for k.*

e. *Each iteration of the while-loop increases the NSW.*

Proof. We prove statements a) to d) by induction on the number of iterations in phase 3. Before the first iteration a) and d) trivially hold. Claim b) holds since in phase 2 we allocate only goods that are light for every agent and since the next good is always added to a lightest bundle. Claim c) holds by Lemma 4.

Assume now that a) to c) hold before the i-th iteration and that we move a good g from A_t to A_k in iteration i. We will show that d) holds for A_t and A_k and that a) to c) hold after iteration i.

By the condition of the while-loop, we have $v_t(A_t) > p \cdot (v_k(A_k) + 1)$. Thus A_t contains only goods that are heavy for t by part b) of the induction hypothesis. Let g be any good in A_t. If we also have $v_k(g) = p$, then moving g from A_t to A_k would result in an allocation of heavy goods that is lexicographically larger, a contradiction to c). Thus $v_k(g) = 1$.

After moving g, c) holds by Lemma 5. Note that g is given from an agent with maximum utility and is not heavy for its new owner.

Since k is the largest index such that $v_k(A_k) = v_1(A_1)$ before the i-th iteration, b) holds after the i-th iteration.

It remains to show that part a) holds after the i-th iteration. The value of the k-th bundle increases by 1 and the value of the t-th bundle decreases by p. We need to show $v_t(A_t) - p \ge v_{t-1}(A_{t-1}) + \delta$, where $\delta = 1$ if $k = t-1$ and $\delta = 0$ otherwise.

- If $k = t-1$, we have $v_t(A_t) \ge p \cdot (v_{t-1}(A_{t-1}) + 1) + 1$ and hence $v_t(A_t) - v_{t-1}(A_{t-1}) - p - 1 \ge (p-1) \cdot v_{t-1}(A_{t-1}) \ge 0$.
- If $k < t-1$, by definition of t, $v_t(A_t) > v_{t-1}(A_{t-1})$. If all goods in A_{t-1} are heavy for $t-1$, the difference in weight is at least p and we are done. If A_{t-1} contains a good that is light for $t-1$, then $v_{t-1}(A_{t-1}) \le v_k(A_k) + 1$ by condition b) and hence $v_t(A_t) \ge p \cdot v_{t-1}(A_{t-1}) + 1$. This implies $v_t(A_t) \ge v_{t-1}(A_{t-1}) + p$ except if $v_{t-1}(A_{t-1}) = 0$. In the latter case, $k = t-1$, but we are in the case $k < t-1$.

  We also need to show that after moving the good, $v_k(A_k) \le v_{k+1}(A_{k+1})$. By the choice of k, $v_k(A_k) \le v_{k+1}(A_{k+1}) + 1$ holds before moving the good. After moving the good, by condition d), $v_k(A_k)$ increases by 1 and therefore, $v_k(A_k) \le v_{k+1}(A_{k+1})$.

Let us finally show e). By d) we know that whenever a good is moved from bundle A_t to bundle A_k, all goods in A_t are heavy for t and light for k. Therefore, moving a good from A_n to A_1 increases the NSW if and only if $(v_n(A_n)-p)(v_1(A_1)+1) > v_n(A_n)v_1(A_1)$ which holds true if and only if $v_n(A_n) > p \cdot v_1(A_1) + p$. Q.E.D.

Lemma 7. *Let A be the output of Algorithm 1. Let OPT be an allocation that maximizes the NSW and, subject to that, maximizes $|OPT^H|$. Then $|OPT^H| \geq |A^H|$.*

Proof. Assume $|\mathrm{OPT}^H| < |A^H|$. By the choice of OPT, A cannot maximize the NSW (since $|\mathrm{OPT}^H|$ is maximum for any allocation maximizing the NSW). We first show that we may assume $\mathrm{OPT}_i^H \subseteq A_i^H$ for all i. By Lemma 6c), A^H is lexmax among all heavy-only allocations of cardinality $|A^H|$. We obtain a lexmax heavy-only allocation C^H of cardinality $|\mathrm{OPT}^H|$ from A^H by repeatedly removing a good from the lowest indexed bundle of maximum utility. Since OPT^H is a lexmax heavy-only allocation of cardinality $|\mathrm{OPT}^H|$, the utility profiles of C^H and OPT^H agree and hence there is a bijection π of the set of agents such that $|C_i^H| = |\mathrm{OPT}_{\pi(i)}^H|$. Let ℓ_i be the number of goods in $\mathrm{OPT}_{\pi(i)}$ which are light for $\pi(i)$. Note that the number of goods which are not allocated under C^H is equal to the number of goods that are light to their owner under OPT, i.e, $\Sigma_{i\in[n]}\ell_i$. Obtain an allocation C from C^H by giving ℓ_i not yet allocated goods to C_i. Then $v_i(C_i) \geq v_{\pi(i)}(OPT_{\pi(i)})$ for all i. Thus, C is optimal and $v_i(C_i) = v_{\pi(i)}(OPT_{\pi(i)})$. Also $C_i^H \subseteq A_i^H$ for all i. So choosing OPT as C, we may assume $\mathrm{OPT}_i^H \subseteq A_i^H$ for all i.

Let t be such that $A_t^H \setminus \mathrm{OPT}_t^H \neq \emptyset$ and let $g \in A_t^H \setminus \mathrm{OPT}_t^H$. In OPT, g is allocated to some agent j as a light good. Note that OPT_t is heavy-only (Otherwise, interchange the light good in OPT_t with g and re-convert g to a heavy good, thus improving the NSW of OPT, a contradiction.), and hence $v_t(\mathrm{OPT}_t) + p \leq v_t(A_t) \leq v_n(A_n)$.

Since A is not optimal there is an agent k such that $v_k(\mathrm{OPT}_k) > v_k(A_k)$. Since $\mathrm{OPT}_k^H \subseteq A_k^H$, the bundle OPT_k contains a good that is light for k.

Since A is the output of Algorithm 1, the while condition of Algorithm 1 (line 10) is violated and hence,

$$v_n(A_n) \leq p \cdot v_1(A_1) + p.$$

Therefore, we get

$$v_t(\mathrm{OPT}_t) + p \leq v_n(A_n) \leq p \cdot v_1(A_1) + p \leq p \cdot v_k(A_k) + p \leq p \cdot v_k(\mathrm{OPT}_k)$$

and hence,

$$v_t(\mathrm{OPT}_t) \leq p \cdot v_k(\mathrm{OPT}_k) - p.$$

Moving a light good from k to j, and taking g from j's bundle and allocating it as a heavy good to t increases the number of heavy goods allocated to t by one. This reallocation of goods does not decrease the NSW as

$$(v_k(\mathrm{OPT}_k)-1)(v_t(\mathrm{OPT}_t)+p)-v_k(\mathrm{OPT}_k)v_t(\mathrm{OPT}_t)=p\cdot v_k(\mathrm{OPT}_k)-v_t(\mathrm{OPT}_t)-p\geq 0.$$

For the new allocation $\widehat{\mathrm{OPT}}$, we have

$$\mathrm{NSW}(\widehat{\mathrm{OPT}})\geq \mathrm{NSW}(\mathrm{OPT}) \qquad \text{and} \qquad |\widehat{\mathrm{OPT}}^H|>|\mathrm{OPT}^H|,$$

a contradiction to the choice of OPT. Q.E.D.

LEMMA 8. *Let $\hat{A}$ be the partial allocation after phase 1 of Algorithm 1. Then $|\hat{A}^H|\geq|OPT^H|$ for any optimal allocation OPT.*

Proof. Assume otherwise. Then there is a heavy edge (i,g) which is not in $\hat{A}$. Allocating g to agent i increases the NSW, a contradiction to Lemma 4. Q.E.D.

By Lemma 7 and Lemma 8, we can assume in some round in phase 3 of Algorithm 1 with allocation $\tilde{A}$, $|\tilde{A}^H|=|\mathrm{OPT}^H|$ for some optimal allocation OPT. Then, by Lemma 2 and Lemma 6.c, we get the following Corollary.

COROLLARY 4. *There is an optimal allocation OPT such that $\tilde{A}^H$ and OPT^H have the same utility profile.*

So far, we have proved that considering only heavy allocated goods in $\tilde{A}$ and OPT, we end up having the same utility profile. Light goods are allocated in a greedy manner during phases 2 and 3 of Algorithm 1. By Lemma 3 we conclude that $\tilde{A}$ is of maximum NSW. In each round of the phase 3 of Algorithm 1, the NSW increases, so $\mathrm{NSW}(A)\geq\mathrm{NSW}(\tilde{A})=\mathrm{NSW}(\mathrm{OPT})$.

THEOREM 1. *There exists a polynomial-time algorithm computing a maximum NSW allocation for integral instances, i.e., when $q=1$ and p is an integer greater than one.*

Proof. We have already proved that the output of Algorithm 1 is an allocation maximizing NSW. It remains to prove that this algorithm runs in polynomial time. Algorithm 2 (i.e., the first phase of Algorithm 1) runs in polynomial time, Barman et al. [10]. The second phase clearly takes polynomial time. By Lemma 6.d, the number of heavy goods under A is decreasing after each iteration of the third phase. Therefore, this phase can be executed at most m times. Overall, Algorithm 1 terminates in polynomial time. Q.E.D.

4. Half-Integral Instances We give a polynomial time algorithm for half-integral instances, i.e., $q = 2$ and p is an odd integer greater than two. We will describe a set of improvement rules that can turn any allocation into an optimal allocation. We will first deal with the case that all heavy goods are allocated as heavy goods (Sections 4.1, 4.2, and 4.3). In the first section, we discuss the improvement rules informally. In Section 4.2 we then introduce the improvement rules formally. As for integral instances, in an optimal allocation OPT only bundles of value x_O, $x_O + {}^1\!/_2$ and $x_O + 1$ can contain light goods; here x_O is the minimal value of any bundle. The first set of improvement rules optimizes the bundles of value larger than $x + 1$. Such bundles are heavy-only. The second set of improvement rules then optimizes the allocation of bundles which may also contain light goods. We will introduce *alternating walks* as a generalization of alternating paths. After the characterization of optimal allocations, we turn to the algorithm in Section 4.3. We establish a connection to matchings with parity constraints, Akiyama et al. [1], and exploit known algorithms for maximum matchings with parity constraints. In Section 4.4 we then deal with the situation that heavy goods can be allocated as light goods. There the approach is similar to the case of integral instances. We first compute an optimal allocation under the constraint that all heavy goods are allocated as heavy goods. Then we repeatedly take a heavy good from a bundle of highest value and allocate it as a light good to a bundle of minimum value. In contrast to integral instances it will be necessary to reoptimize after each such move.

We will use s as a short-hand for p/q. Recall that $q = 2$ and p is an odd integer greater than two.

4.1. Improvement Rules As explained in the introduction, our goal is to identify improvement rules that transform any suboptimal allocation into an optimal allocation. So if A is a suboptimal allocation, one of the rules applies and improves the NSW of A. In this section, we give an informal introduction to our collection of improvement rules. For this discussion we assume that all heavy goods are allocated as heavy goods, and that ${}^p\!/_q = {}^3\!/_2$. Instead of using ${}^p\!/_q$, we sometimes use s.

Example 3. Consider a light good. A light good can be given to any agent. So if one has two bundles of values x and y with $x < y$, and an unallocated light good, one should allocate the good to the lighter bundle, since $(x+1)/x > (y+1)/y$ and hence allocating the light good to the lighter bundle leads to a greater increase in NSW. This rule is called *greedy allocation of light goods*. Another way of stating this rule is: If x is the minimum value of any bundle and there is a bundle of value larger than $x + 1$ containing a light good, move the light good to the bundle of value x. ■

Example 4. We turn to the allocation of heavy goods and the interaction between heavy and light goods. Assume we have two agents i and j owning bundles of value x and $x + 1$, respectively, and i likes[2] a heavy good in j's bundle and owns a light good. Then moving a heavy good from j to i and a light good from i to j would give both bundles value $x + 1/2$ and improve the NSW. The connection between i and j does not have to be direct, but can go through an alternating path as the following example shows:

$$i \overset{\bar{A}}{\text{---}} g \overset{A}{\text{---}} u \overset{\bar{A}}{\text{---}} g' \overset{A}{\text{---}} j.$$

In this example, i owns a light good and likes a heavy good g which u owns, who in turn likes a heavy good g' owned by j. In this diagram, an edge between an agent and a good indicates that the good is heavy for the agent. The superscript A indicates that the good is allocated to the agent, the superscript $\bar{A}$ indicates that the good is not allocated to the agent. We move g' from j to u and g from u to i and move a light good from i to j. The change in the allocation of heavy goods is akin to augmenting the path from i to j to A^H; here A^H denotes the allocation of the heavy goods in A. Of course, the path from i to j might have more than one intermediate agent. ■

Augmenting alternating paths is powerful, but not enough. We give four examples to illustrate this point. The first three examples re-illustrate that it is advantageous to have bundles of value $x + 1$ containing a light good. In Example 7, we even make a preparatory move to create such a bundle.

Example 5. Assume agents i and j own bundles of value x and agent h owns a bundle of value $x + 1$ containing a light good. i owns a light good and likes a heavy good in j's bundle. Assume i takes the heavy good from j in return for a light good and j gets another light good from h. There are three agents involved in the update. Before the update, i and j own bundles of value x and h owns a bundle of value $x + 1$. After the update, we have two bundles of value $x + 1/2$ and one bundle of value x. There is another way to interpret this change. We first make a move that does not change NSW: we move a light good from h to j turning h into a bundle of value x and j into a bundle of value $x + 1$. In a second step, we have the exchange between i and j; i receives a heavy good from j in return for a light good. We will say that h plays the role of a *facilitator* for the transformation, i.e., h owns a bundle of value $x + 1$ containing a light good and gives up the light good so that a transformation becomes possible. ■

Example 6. We can also start with two bundles of value $x + 1$ and one bundle of value x. One of the bundles of value $x + 1$ contains two light goods, and its owner likes a heavy good in the other bundle of value $x + 1$. He takes the heavy good in return for a light good and gives the other light good to a bundle of value x. As a consequence, both bundles of value $x + 1$ turn into bundles of value $x + 1/2$. The bundle of value x becomes a bundle of value $x + 1$. ■

[2] Instead of saying that an agent consider a good heavy we also say that the agent likes the good.

Example 7. We have four bundles with values x, x, $x + 1/2$ and $x + 1$, respectively. One of the bundles of value x likes a heavy good in the other bundle of value x and owns a light good. We would be in the situation of the previous example, if the bundle of value $x + 1$ contains a light good. Assume it does not, but the bundle of value $x + 1/2$ likes a heavy good in the bundle of value $x + 1$ and contains two light goods. It pulls a heavy good from the bundle of value $x + 1$ in return for a light good. So the two bundles swap values and the utility profile does not change. However, we now have a bundle of value $x + 1$ containing a light good. Such bundles facilitate transactions. ■

Example 8. The final example shows that we need structures that go beyond augmenting paths. Consider the following structure

$$i \overset{A}{—} g_1 \overset{\bar{A}}{—} h \overset{\bar{A}}{—} g_2 \overset{A}{—} j$$

and assume that h owns three light goods and i and j own bundles of value x and $x+1$ respectively. Note that h is interested in g_1 and g_2. Then h gives two light goods to i and one light good to j and we change the heavy part of the allocation to

$$i \overset{\bar{A}}{—} g_1 \overset{A}{—} h \overset{A}{—} g_2 \overset{\bar{A}}{—} j.$$

Note that the value of h's bundle does not change; h gives away three light goods and obtains two heavy goods; i and j now own bundles of value $x + 1/2$. ■

4.2. All Heavy Goods are Allocated as Heavy Goods Throughout this section we assume that all heavy goods are allocated as heavy. We drop this assumption in Section 4.4. When we refer to the value of a bundle we mean the value to its owner. A is our current allocation which the algorithm tries to change into an optimal allocation by the application of improvement rules, O is either any optimal allocation or an optimal allocation closest to A in a sense to be made precise below. We use x_A or simply x to denote the minimum value of any bundle in A and x_O for the minimum value of any bundle in O. We use A_d and O_d to denote the set of agents that own a bundle of value $x_A + d$ in A or O, respectively. Note that we use the reference value x_A also for O. We will use this notation only with $d \in \{-1/2, 0, 1/2, 1, 3/2\}$. We will use the short-hand *"a bundle in A_d"* for a bundle owned by an agent in A_d.

A_i is the bundle owned by agent i. We use h, i, j and sometimes u and v to denote agents and g and g' to denote goods. Remember v_i is the valuation function of agent i and we define $w_i := v_i(A_i)$ to be the value of i's bundle for i in N. The *heavy value* of a bundle is the value of the heavy goods contained in the bundle. We say that a bundle is *heavy-only* if all goods in the bundle are heavy for its owner. A bundle of value $x + 1$ containing a light good is called a *facilitator*.

Our goal is to transform A into an optimal allocation by the application of improvement rules. Each improvement lexicographically increases the potential function:

$$\Phi = (\text{NSW}(A),\ \text{number of agents in } A_1 \text{ owning a light good}).$$

Let G^H be the bipartite graph with agents on one side and heavy goods on the other side. There is an edge connecting an agent i and a good g, if g is heavy for i. We use A^H and O^H to denote the heavy part of the allocations A and O. They are subsets of edges of G^H. The allocation O is closest to A if $A^H \oplus O^H$ has minimum cardinality among all optimal allocations.

The basic improvement rules are based on alternating paths. We will consider two kinds of alternating paths in G^H: A-$\bar{A}$-alternating paths and A-O-alternating paths. In an A-B-alternating path, the edges in $A^H \setminus B^H$ and $B^H \setminus A^H$ alternate. We will use A-$\bar{A}$-alternating paths in the algorithm and A-O-alternating paths for showing that some improvement rule applies to any suboptimal A.

Let i and j be agents. An *A-B-alternating path from i to j* is an A-B-alternating path with endpoints i and j in which i is incident to an edge in A (and hence j is incident to an edge in B). So a B-A-alternating path from i to j uses a B-edge incident to i.

4.2.1. Basic Improvement Rules

Lemma 9. *Let A be any allocation and let x be the minimum value of any bundle in A. Let i be any agent. For parts* b) to g)*, let j be any other agent, and let P be an A-$\bar{A}$-alternating path from i to j.*

a) If $w_i > x + 1$ and A_i contains a light good, moving the light good to a bundle of value x improves the NSW of A.
b) If $w_i \geq w_j + \lceil s \rceil$, augmenting P to A improves the NSW of A.
c) If $w_i \geq w_j + 1$ and A_j contains more than $s - (w_i - w_j)$ light goods, augmenting P to A and moving $\max(0, \lceil s - (w_i - w_j) + 1/2 \rceil)$ light goods from j to i improves the NSW of A.
d) If $w_i \in \{x+1, x+3/2\}$, $w_j = x+1$, and A_j contains $\lceil s \rceil$ light goods, augmenting P to A and moving $\lfloor s \rfloor$ light goods from A_j to A_i and another light good from A_j to any bundle of value x improves the NSW of A.
e) If i owns at least two heavy goods more than j and $w_i \geq x + 3/2$, one of the cases b), c), or d) applies and the NSW of A can be improved.
f) If $w_i = x$, $w_j = x+1$, and A_j contains $\lceil s \rceil$ light goods, augmenting P to A and moving $\lceil s \rceil$ light goods from A_j to A_i improves the NSW of A.
g) If $w_i = x+1$, $w_j = x + 1/2$, A_i is heavy-only, and A_j contains $\lceil s \rceil$ light goods, augmenting P to A and moving $\lfloor s \rfloor$ light goods from A_j to A_i leaves the NSW of A unchanged and increases the number of bundles of value $x+1$ containing a light good.

h) For a fixed agent i, one can determine in time $O(m)$, whether any of the improvement rules is applicable. For all agents, one can do so in time $O(nm)$.

Proof.

a) The sum of the values of the two bundles does not change and the new values lie strictly inside the interval defined by the old values. The claim follows from Lemma 1a).

b) The sum of the values of the two bundles does not change and the new values lie strictly in the interval defined by the old values. The claim follows from Lemma 1a).

c) If $w_i \ge w_j + \lceil s \rceil$, the claim follows from part b). So assume $w_j + 1 \le w_i \le w_j + s$. By assumption, A_j contains at least $r = \lceil s - (w_i - w_j) + 1/2 \rceil$ light goods. After augmenting P and moving r light goods from A_j to A_i, the value of A_i is $w_i - s + r > w_i - s + (s - (w_i - w_j)) = w_j$ and similarly the value of A_j is less than w_i. Thus the NSW increases by Lemma 1a).

d) Before the augmentation, we have bundles of value $x+1+d$ with $d \in \{0, 1/2\}$, $x+1$ and x. After the augmentation, we have bundles of value $x+1+d-s+\lfloor s \rfloor = x + 1/2 + d$, $x+1+s-\lceil s \rceil = x + 1/2$ and $x+1$. By Lemma 1a), $(x+1+d)x > (x+1/2+d)(x+1/2)$. Thus the NSW improves.

e) The heavy value of j is at most $w_i - 2s$. If j is heavy-only, augmenting P improves A according to part b). If j owns a light good and part a) does not apply, $w_j \le x+1 < w_i$ and j owns at least $w_j - (w_i - 2s) = 2s - (w_i - w_j)$ light goods. If $w_i \ge w_j + 1$, A can be improved according to part c). Otherwise, we have $w_i = x + 3/2$ and $w_j = x+1$, and A can be improved according to part d).

f) Before the augmentation, we have bundles of value x and $x+1$, after augmentation we have two bundles of value $x + 1/2$. The NSW improves by Lemma 1a).

g) Before the augmentation we have bundles of value $x+1$ and $x + 1/2$ and after the augmentation we have bundles of value $x+1-s+\lfloor s \rfloor = x + 1/2$ and $x + 1/2 + s - \lfloor s \rfloor = x+1$. Thus the NSW does not change. The bundle A_j now has value $x+1$ and contains a light good.

h) A depth first search for an augmenting path starting in an agent i takes linear time $O(m)$.

4.2.2. Range Reduction We need a finer distinction of the rules in Lemma 9. Let $x = x_A$ be the minimum value of a bundle in A and let k_0 be minimal such that $k_0 s > x+1$. We call the rules a) to e) when applied with an agent i of value larger than $x+1$ *reduction rules*. An allocation A is *reduced* if no reduction rule applies to it. *Throughout this section, A is a reduced allocation and O is an optimal allocation closest to A.* We will show that the bundles of value ks in A and O are identical for all $k \ge k_0$. This will allow us to restrict attention to the bundles of value x, $x + 1/2$, and $x+1$ in A and to the bundles of value x, $x \pm 1/2$, and $x+1$ in O.

Observation 6. $A^H \oplus O^H$ is acyclic.

Proof. Assume otherwise. Let C be a cycle in $A^H \oplus O^H$. Replacing O^H by $O^H \oplus C$ and leaving the allocation of the light goods unchanged yields an optimal allocation closer to A.

Q.E.D.

We *decompose* $D := A^H \oplus O^H$ into maximal alternating paths. Goods have degree zero or two in D. For a good of degree zero, there is no path using it. For a good of degree two, there is one alternating path passing through it. For an agent i, let $hdeg_i^A$ ($hdeg_i^O$) be the number of A-edges (O-edges) incident to i in A^H (O^H). For i, we form $\min(hdeg_i^A, hdeg_i^O) - |A_i^H \cap O_i^H|$ pairs of A- and O-edges incident to i. Then $\max(hdeg_i^A, hdeg_i^O) - \min(hdeg_i^A, hdeg_i^O)$ alternating paths start in i. Depending on which degree is larger, the paths start with an A- or an O-edge. The decomposition of $A^H \oplus O^H$ into alternating paths needs not be unique.

Recall that N denotes the set of agents.

Lemma 10. *Let A be reduced and let O be an optimal allocation closest to A. Let k_0 be minimal with $k_0 s > x+1$. For $k \ge k_0$ let R_k be the set of agents that own a bundle of value ks in A and let $S_k = \cup_{j \ge k} R_j$ be the agents that own a bundle of value at least ks in A.*

Agents in R_k own k heavy goods and no light good in A. O agrees with A on S_{k_0}, i.e., $A_i = O_i$ for all $i \in S_{k_0}$. Moreover, in A and O each agent in $N \setminus S_{k_0}$ owns at most $k_0 - 1$ heavy goods. Finally, $x_O + 1 \le k_0 s$.

Proof. We use downward induction on k to show that O and A agree on S_k. Assume that they agree on S_{k+1}; then agents in S_{k+1} have degree zero in $A^H \oplus O^H$. Since S_{k+1} is empty for large enough k, the induction hypothesis holds for large enough k. Let

$$R'_k := \{j \in N \mid j \notin S_{k+1} \text{ and there is an } A\text{-}O\text{-alternating path from } i \in R_k \text{ to } j\}.$$

We will establish a sequence of Claims and then complete the proof of the Lemma.

Claim 4. $x_O < x + s$.

Proof. Let L_0 be the set of agents owning a light item in A and let L be all agents that can be reached from an agent in L_0 by a $\bar{A}$-A-alternating path. Then

a) Agents in L_0 have value in $\{x, x+1/2, x+1\}$. Otherwise, Lemma 9a) allows to improve A.
b) All agents in L have value at most $x+s$. Assume otherwise. Then there is an agent of value $x + \lceil s \rceil$ or more that can be reached from an agent of value at most $x+1$ owning a light item and Lemma 9c) allows to improve A.
c) All heavy goods that are liked by an agent in L are also owned by an agent in L. Assume otherwise, say good g is liked by $j \in L$, but owned by $h \notin L$. We could then extend the alternating path ending in j by $j \overset{\bar{A}}{—} g \overset{A}{—} h$ and put h into L.

The total value of the goods assigned to the agents in L by O cannot be larger than the total value in A and hence the average value of an agent in L is no larger in O than in A. This holds since in A all light goods are owned by agents in L_0, $L_0 \subseteq L$, and all heavy goods that are liked by an agent in L are owned by an agent in L. So O cannot assign any additional value to the agents in L. The average value of a bundle in L in allocation A is strictly less than $x+s$. Thus $x_O < x+s$. Q.E.D.

Claim 5. *In A, each bundle in $R'_k \setminus R_k$ contains exactly $k-1$ heavy goods.*

Proof. By definition, bundles in $N \setminus S_{k+1}$ contain at most k heavy goods in A. Bundles that contain exactly k heavy goods belong to R_k and hence the bundles in $R'_k \setminus R_k$ contain at most $k-1$ heavy goods. Each $j \in R'_k \setminus R_k$ is reachable by an A-O-alternating path from an $i \in R_k$ and $w_i = ks > x+1$. If j does not contain $k-1$ heavy goods, Lemma 9e) implies that A is not reduced. Thus, j contains exactly $k-1$ heavy goods. Q.E.D.

Claim 6. *In O, each bundle in $N \setminus S_{k+1}$ contains at most k heavy goods.*

Proof. Assume that there is an agent i in $N \setminus S_{k+1}$ that owns $k+1$ or more heavy goods in O. Then $w_i^O \ge ks + s \ge x + 3/2 + s \ge x_O + 3/2$, where the last inequality comes from Claim 4. Since i owns more heavy goods in O than in A, there is an O-A-alternating path starting in i. Consider a maximal such path and let j be the other endpoint. Then j owns fewer heavy goods in O than in A. Since we know already that A and O agree on S_{k+1}, j contains at most $k-1$ heavy goods in O, a contradiction to the optimality of O (Lemma 9e). Q.E.D.

Claim 7. *Heavy goods assigned to agents in R'_k by A are also assigned to agents in R'_k by O.*

Proof. Let g be any good that A assigns to an agent in R'_k, say j, and let h be the owner of g in O. We need to show $h \in R'_k$. This is obvious if $h = j$. So assume otherwise. Since A and O agree on S_{k+1}, $h \notin S_{k+1}$. Let P be an A-O-alternating path from $i \in R_k$ to j; $i = j$ is possible. We extend P by $j \overset{A}{—} g \overset{O}{—} h$ and hence h can be reached by an alternating path starting with an A-edge from $i \in R_k$. Thus, $h \in R'_k$. Q.E.D.

Claim 8. *O assigns k heavy goods to at least $|R_k|$ agents in R'_k.*

Proof. In A, the agents in R_k own k heavy goods (by definition) and the agents in $R'_k \setminus R_k$ own $k-1$ heavy goods (Claim 5). So the number of heavy goods allocated by A to the agents in R'_k is $m_k^A := (k-1)|R'_k| + |R_k|$. All heavy goods assigned to agents in R'_k by A are also assigned to them by O (Claim 7) and no agent in R'_k is assigned more than k heavy goods in O (Claim 6). Let m_k^O be the number of agents in R'_k to which O assigns k heavy goods. Then $m_k^O k + (|R'_k| - m_k^O)(k-1) \ge m_k^A$ and hence $m_k^O + |R'_k|(k-1) \ge (k-1)|R'_k| + |R_k|$. Thus $m_k^O \ge |R_k|$. Q.E.D.

Claim 9. *$x_O + 1 \le ks$. Bundles in O of heavy value ks do not contain a light good.*

Proof. Let L be the set of agents that own bundles of value less than ks in A. Then $L = N \setminus S_k$ and A assigns all light goods to the agents in L. In A, there is at least one bundle of value x in L. So the average value of a bundle in L is less than $ks - 1/2$ in A.

We know already that O and A agree on S_{k+1}, and that O assigns at least k heavy goods to at least $|R_k|$ many agents in R'_k (by Claim 8). Choose any $|R_k|$ of them, and let L' be the remaining agents. Then $|L'| = |L|$ and the total value of the O-bundles of the agents in L' is at most the total value of the A-bundles of the agents in L since the number of heavy goods assigned to them cannot be larger and all light goods are assigned to agents in L by A. Thus their average value is less than $ks - 1/2$ and hence $x_O < ks - 1/2$.

A bundle of heavy value ks containing a light good would have value larger than $x_O + 1$, a contradiction to the optimality of O. Q.E.D.

Claim 10. *Agents in R_k own k heavy goods in O.*

Proof. Let i be an agent in R_k. Then $hdeg_i^O \le k = hdeg_i^A$; the equality holds by the definition of R_k and the inequality holds by Claim 6. Assume for the sake of contradiction, that there is an agent $i \in R_k$ with $k = hdeg_i^A > hdeg_i^O$. Consider an A-O-alternating path (in the alternating path decomposition) starting in i. Let j be the other end of the path. Then, $hdeg_j^O > hdeg_j^A$. Also, $j \in R'_k$ and hence $hdeg_j^A \ge k - 1$ (Claim 5). Since $hdeg_j^O \le k$ (Claim 5), we have $hdeg_j^O = k$ and $hdeg_j^A = k - 1$. Therefore, the value of j in O is at least ks and $j \in R'_k \setminus R_k$.

If $hdeg_i^O = k - 1$, we augment P to O and also exchange the light goods (if any); the utility profile of O does not change and O moves closer to A, a contradiction to the choice of O.

So $hdeg_i^O \le k - 2$, and then the heavy value of i in O is at most $ks - 2s$ and hence i owns at least $w_i^O - (ks - 2s) \ge 2s - (w_j^O - w_i^O) > s - (w_j^O - w_i^O)$ light goods. If the value of i is no larger than $ks - 1$, O can be improved (Lemma 9c)).

So $w_i^O \ge ks - 1/2$ and hence i contains at least $2s$ light goods in O (since it heavy value in O is at most $ks - 2s$). Thus $w_i^O \le x_O + 1$ and hence $x_O \ge ks - 1/2 - 1 = ks - 3/2$. The heavy degree of i in A is at least two more than the heavy degree of i in O. Therefore there is a second A-O-alternating path starting in i, say Q. It ends in a node h. As above for j, we conclude $hdeg_h^O = k$ and $hdeg_h^A = k - 1$. Then, $j \ne h$ as only one alternating path can end in j as well as in h. It is however possible, that j lies on the path from i to h or that h lies on the path from i to j.

We have $w_j^O \ge ks$ and $w_h^O \ge ks$. We augment P and Q to O and use the $2s$ light goods on i as follows: We give $\lfloor s \rfloor$ light goods to each of j and h, and one to a bundle of value x_O. The value of i does not change, the values of j and h go down by $1/2$ each and the value of an x_O bundle goes up by one. We may assume $w_j^O \ge w_h^O$. Since $w_j^O \ge ks \ge x_O + 1$ and $w_h^O \ge ks \ge x_O + 1$ by Claim 9, we can apply Lemma 1b) with $a := w^O_j$, $b := w^O_h$, $c := x_O$, $d_1 := 1/2$ and $d_2 := 1/2$, a contradiction to the optimality of O. Q.E.D.

CLAIM 11. *O agrees with A on R_k, i.e., $O_i = A_i$ for all $i \in R_k$. Bundles owned by agents in $N \setminus S_k$ do not contain k heavy goods in either A or O.*

Proof. Consider the path decomposition of $A^H \oplus O^H$. We first establish that there can be no O-A alternating path from an agent j to an agent h with $hdeg^A_j < hdeg^O_j = k$ and $hdeg^O_h < hdeg^A_h = k-1$. Assume otherwise and let P be such a path. It uses an O-edge incident to j and an A-edge incident to h. Also O_j contains no light good by Claim 9.

By assumption, $hdeg^O_h \le k-2$ and hence the heavy value of h is at most $ks - 2s$. If the value of h is no larger than $ks - 1$, we augment P to O and possibly move light goods from h to j; note that h contains $w^O_h - (ks - 2s)$ light goods. This is Lemma 9c). This improves O, a contradiction.

So $w^O_h \ge ks - 1/2$ and hence O_h contains at least $2s$ light goods. Thus $w^O_h \le x_O + 1$ and hence $x_O \ge ks - 1/2 - 1 = ks - 3/2$. If $w^O_h = ks - 1/2$ we augment the path to O and move $\lfloor s \rfloor$ light goods from h to j. This does not change the utility profile of O and moves O closer to A.

We cannot have $w^O_h > ks$ as this would imply $x_O + 1 > ks$, a contradiction to Claim 9.

This leaves the case $w^O_h = ks$. Then $ks = x_O + 1$, since $w^O_h \le x_O + 1$ as shown above and $ks \ge x_O + 1$ by Claim 9. Also h owns at least $2s$ light goods. We augment the path to O and move $\lceil s \rceil$ light goods from h, $\lfloor s \rfloor$ of them to j and one of them to a bundle of value x_O. Before the augmentation we have two bundles of value ks and one bundle of value x_O, after the change, we have two bundles of value $ks - 1/2$ and one bundle of value $x_O + 1$. Since $ks = x_O + 1$ (Lemma 9d)), the NSW improves, a contradiction.

We can now proceed to the Claim proper.

By Claims 5 to 10, agents in R_k own exactly k heavy goods in A as well as in O. If O does not agree with A on R_k, there is an A-O-alternating path P in the path decomposition of $O^H \oplus A^H$ passing through an agent $i \in R_k$. Let j and h be the endpoints of the path. Say j is incident to an O-edge and h is incident to an A-edge. Thus $j \in R'_k$. Since $hdeg^O_j > hdeg^A_j \ge k-1$ by Claim 10, we have $hdeg^A_j = k-1$ and $hdeg^O_j = k$. Since h is incident to an A-edge, $hdeg^O_h < hdeg^A_h$ and hence $hdeg^A_h \le k-1$ since $hdeg^A_h = k$ implies $hdeg^O_h = k$ by the preceding claim.

But, as shown above, such a path P does not exist and hence $O^H_i = A^H_i$ for all $i \in R_k$; A_i does not contain any light good. O_i does neither since it would have value at least $k_0 s + 1 > x_O + 1$ (by Claim 9) otherwise. Thus, $O_i = A_i$ for all $i \in R_k$.

Assume finally that there is a bundle O_j with $j \notin S_k$ containing k heavy goods. Since $j \notin S_k$, A_k contains fewer than k heavy goods. So there is an O-A-alternating path P starting in j. Let h be the other endpoint. The path ends with an A-edge incident to h and hence there are more A-edges than O-edges incident to h. We do not have $h \in S_k$ because this would imply $hdeg_h^O = hdeg_h^A$. Thus, $hdeg_h^A \le k-1$ and $hdeg_h^O \le k-2$. But such a path does not exist and hence bundles O_j with $j \notin S_k$ contain less than k heavy goods. Q.E.D.

We can now complete the proof of Lemma 10. The induction shows that A and O agree on S_{k_0}. Also agents in $N \setminus S_{k_0}$ own at most $k_0 - 1$ heavy goods in A by definition of S_{k_0} and in O by Claim 11. Finally, $x_O + 1 \le k_0 s$ by Claim 9. Q.E.D.

At this point, we know that $A_i = O_i$ for all $i \in S_{k_0}$, where k_0 is minimal with $k_0 s > x+1$ and S_{k_0} is the set of agents that own bundles of value at least $k_0 s$ in A. We may therefore remove the agents in S_{k_0} and their bundles from further consideration. We call A and O *shrunken* after this reduction. The remaining bundles have value x, $x + 1/2$, and $x+1$ in A and value x_O, $x_O + 1/2$, ... in O; in Lemma 15, we will show an upper of $x+1$ on the value of any bundle in O. Moreover, the remaining bundles contain at most $k_0 - 1$ heavy goods in both A and O, and $x_O \le x + 1/2$ since the average value of a bundle in $A_{low} = A_0 \cup A_{1/2} \cup A_1$ is strictly less than $x+1$ as there is at least one bundle of value x and the average for O is the same.

Lemma 11. *Let A be reduced and let O be an optimal allocation closest to A. Then O consists of A restricted to the agents in S_{k_0} and an optimal allocation for the bundles in $A_{low} = A_0 \cup A_{1/2} \cup A_1$.*

Proof. By Lemma 10, $A_i = O_i$ for all $i \in S_{k_0}$. The remaining bundles in A belong to A_{low}. O allocates the goods in these bundles optimally to the agents in A_{low}. Q.E.D.

Lemma 12. *A reduced and shrunken allocation can be constructed in time $O(nm^3)$.*

Proof. We initialize A by allocating each heavy good to an agent that considers it heavy and then adding the light items greedily, i.e., we iteratively add the light items to the least valued bundle. We then apply the reduction rules. For each agent i with $w_i \ge x + 3/2$, we search for an A-$\bar{A}$ alternating path satisfying one of the cases a) to e) of Lemma 9. Whenever we find an improving path, we apply it. Each search for an improving path takes time $O(nm)$.

In order to bound the number of augmentations consider the potential $\sum_i w_i^2$. The potential is non-negative, is at most m^2, and decreases by at least $1/2$ in each augmentation. For example, rule d) converts bundles of value $x + 3/2$, $x+1$ and x into bundles of value $x+1$, $x+1/2$, and $x+1$ and decreases the potential by 1. Thus we have at most $O(m^2)$ augmentations. Q.E.D.

4.2.3. Only Bundles of Value x, $x+1/2$, and $x+1$ in A

We derive further properties of optimal allocations for A_{low} and then introduce an additional reduction rule.

Further Properties of Optimal Allocations.

LEMMA 13. *Let A be reduced and shrunken. If there is an optimal allocation having at least as many bundles of value $x+1$ as A does, A is optimal. In particular, if A has no bundles of value $x+1$, A is optimal.*

Proof. For $d \in \{0, 1/2, 1\}$, let n_d be the number of bundles of value $x+d$ in A. Then $n_1 = n - n_0 - n_{1/2}$ and the total value is $xn + n_{1/2}/2 + n_1$. Split all goods into portions of $1/2$ (i.e. a light good becomes two portions and a heavy good becomes $2s$ portions for a total of $2xn + n_{1/2} + 2n_1$ portions) and allow these portions to be allocated freely, i.e., any portion can be assigned to any agent, subject to the constraint that there are at least n_1 bundles of value $x+1$. Consider an optimal allocation O satisfying this constraint. In O, we have $n_1' \geq n_1$ bundles of value $x+1$ and at least one bundle of value less than $x+1$ as the average value of a bundle in A is less than $x+1$. Also, if we have bundles of value y and y' with $y' \geq y+1$ in O, we can replace them by bundles of value $y + 1/2$ and $y' - 1/2$ and improve the NSW except if $x+1 \in \{y, y'\}$ and there are exactly n_1 bundles of value $x+1$.

Thus, if $n_1' > n_1$, all bundles have value $x+1/2$ and $x+1$. This is impossible since $n_1'(x+1) + (n - n_1')(x+1/2) - (nx + n_{1/2}/2 + n_1) = n_1' + (n-n_1')/2 - n_{1/2}/2 - n_1 = (n_1' - n_1 + n - n_1 - n_{1/2})/2 > 0$.

On the other hand, if $n_1' = n_1$, all bundles have value z, $z+1/2$ and $x+1$ for some $z \leq x$. If $z < x$, the average value of the bundles of value z and $z+1/2$ in O is less than x. However, the bundles of value x and $x+1/2$ in A have average value at least x. Thus $z = x$, and the number of bundles in O of value x and $x+1/2$ are n_0 and $n_{1/2}$, respectively. Thus A is optimal. Q.E.D.

LEMMA 14. *Let A be reduced and shrunken and let O be optimal and closest to A. Then $x_O \geq x - 1/2$.*

Proof. Assume otherwise, i.e., $x_O \leq x-1$. In O, all light goods are contained in bundles of value at most $x_O + 1$, and hence bundles of value larger than x are heavy-only. Any bundle contains at most (k_0-1) heavy goods (Lemma 10) and hence has heavy value at most $(k_0-1)s$. Since k_0 is minimal with $k_0 s > x+1$, we have $(k_0-1)s \leq x+1$.

If $(k_0-1)s \leq x$, the average value of a bundle in O is strictly less than x (there is a bundle of value x_O and all bundles have value at most x), but the average value of a bundle in A is at least x, a contradiction.

Let $y = (k_0-1)s$ and assume $y \in \{x+1/2, x+1\}$. In O, bundles of value y are heavy-only. Let S be the set of owners of the bundles of value y in O. If their bundles in A have value y or more, the average value of a bundle in A is larger than the average value in O, a contradiction. Note that bundles in $N \setminus S$ have value at least x in A, have value at most x in O, and there is a bundle of value x_O in O.

So there is an agent $i \in S$ whose bundle in A has value less than y. Then $hdeg_i^A \le (k_0 - 2) < hdeg_i^O$. Consider an O-A-alternating path starting in i. It ends in a node j with $hdeg_j^O < hdeg_j^A$. Then $hdeg_j^O \le k_0 - 2$ and hence the value of O_j is at most x. The bundle O_j contains at least $\lceil w_j^O - (k_0-2)s \rceil$ light goods since its heavy value is at most $(k_0-2)s$. We augment the path to O and move $\lceil w_j^O - (k_0-2)s \rceil$ light goods from j to i. Let $\delta \in \{0, 1/2\}$ be such that $\lceil w_j^O - (k_0-2)s \rceil = w_j^O - (k_0-2)s + \delta$. Then the new values of bundles O_i and O_j are $w_j^O + \delta$ and $w_i^O - \delta$. Thus either the utility profile of O does not change (if $w_i^O = w_j^O + 1/2$) and O moves closer to A or the NSW of O improves (if $w_i^O \ge w_j^O + 1$), a contradiction. Q.E.D.

Lemma 15. *Let A be reduced and shrunken and let O be optimal and closest to A. There is no bundle of value more than $x+1$ in O.*

Proof. The average value of a bundle in A is less than $x+1$. Thus $x_O < x+1$, as otherwise the average value of a bundle in O would be at least $x+1$. Assume there is a bundle of value $x+3/2$ or more in O. The bundle contains at most $k_0 - 1$ heavy goods and hence its heavy value is at most $x+1$. So it contains a light good and hence $x_O = x + 1/2$, and the bundle under consideration has value $x+3/2$. The bundles in O have values in $\{x+1/2, x+1, x+3/2\}$. Any bundle of value $x+1/2$ can be turned into a bundle of value $x+3/2$ by moving a light good to it from a bundle of value $x+3/2$.

Recall that O_d denotes the set of agents owning bundles of value $x+d$ in O. If $A_1 \subseteq O_1$, A is optimal by Lemma 13 and hence $A = O$ since O is optimal and closest to A. So there is an i in $(O_{1/2} \cup O_{3/2}) \cap A_1$. Then the parities of $hdeg_i^A$ and $hdeg_i^O$ differ since the weights of the bundles differ by $1/2$. By the first paragraph, we may assume $i \in O_{3/2}$.

Assume first that $hdeg_i^O > hdeg_i^A$. Then there exists an O-A-alternating path starting in i. The path ends in j with $hdeg_j^O < hdeg_j^A \le k_0 - 1$. The heavy value of O_j is at most $(k_0-2)s$ which is at most $x+1-s$. Since the value of O_j is at least x_O, O_j contains at least $x_O - (x+1-s) = s - 1/2 = \lfloor s \rfloor$ light goods. If $w_j^O = x+1/2$, we augment the path, move $\lfloor s \rfloor$ light goods from j to i and improve O, a contradiction, by Lemma 9.c). If $w_j^O = x+1$, we augment the path and move $\lfloor s \rfloor$ light goods from j to i. This does not change the utility profile of O and moves O closer to A, a contradiction. If $w_j^O = x+3/2$, j contains at least $\lceil s \rceil$ light goods. We augment the path, move $\lfloor s \rfloor$ light goods from j to i and one light good from j to a bundle of value x_O. This improves O, a contradiction, by Lemma 9.d).

Assume next that $hdeg_i^O < hdeg_i^A \le k_0 - 1$. Then there exists an A-O-alternating path starting in i. The path ends in j with $hdeg_j^A < hdeg_j^O \le k_0 - 1$. Since O_i has value $x+3/2$ and $hdeg_i^O \le k_0 - 2$, O_i contains at least $x + 3/2 - (x+1-s) = \lceil s \rceil$ light goods. We augment the path to O and remove $\lceil s \rceil$ light goods from O_i. So the value of O_i becomes $x+1$. If O_j has value $x+3/2$, we put $\lfloor s \rfloor$ light goods on j and one light good on a bundle of value x_O. If O_j has value $x+1$ or $x+1/2$, we put $\lceil s \rceil$ light goods on j. This either improves O or does not change the utility profile of O and moves O closer to A, a contradiction. Q.E.D.

In the technical introduction (Section 4.1) we pointed to the importance of bundles of value $x+1$ containing a light good. The following lemma formalizes this observation.

LEMMA 16. *Let A be reduced and shrunken and assume further that Lemma 9.g) is not applicable. Let O be optimal and closest to A, and consider an agent $i \in A_1$. If A_i is heavy-only, O_i is heavy-only and has value $x+1$. If all bundles in A_1 are heavy-only, A is optimal.*

Proof. Consider a heavy-only bundle A_i of value $x+1$. Then $hdeg_i^A = k_0 - 1$ and $x+1 = (k_0-1)s$. Assume for the sake of a contradiction that O_i has either value less than $x+1$ or is not heavy-only. In either case, $hdeg_i^A > hdeg_i^O$ and hence O_i contains at most $k_0 - 2$ heavy goods and thus at least $w_i^O - (x+1-s) = w_i^O - x - 1 + s$ light goods. If $i \in O_0 \cup O_1$, O_i contains at most $k_0 - 3$ heavy goods as the parity of the number of heavy goods is the same as for A_i. This holds since the value of A_i and O_i differ by an integer, namely either zero or one.

Consider an A-O-alternating path starting in i and let j be the other end of the path. Then $hdeg_j^A < hdeg_j^O$ and hence A_j contains at most $k_0 - 2$ heavy goods; its heavy value is therefore at most $x+1-s$. Since the value of A_j is at least x, A_j contains at least $\lfloor s \rfloor$ light goods. At least $\lceil s \rceil$, if A_j has value $x+1$.

By Lemma 9c) $w_j^A \neq x$, by part d) $w_j^A \neq x+1$. So $w_j^A = x + 1/2$. Then by part g), A_j contains less than $\lceil s \rceil$ light goods and hence contains at least $k_0 - 2$ heavy goods.

So we are left with the case where A_j has value $x + 1/2$ and contains $k_0 - 2$ heavy goods. Then O_j contains $k_0 - 1$ heavy goods and hence is a heavy-only bundle of value $x+1$. If the value of O_i is $x - 1/2$, O_i contains at least $s - 3/2$ light goods. We augment the path to O and move $s - 3/2$ light goods from i to j. This does not change the utility profile of O and moves O closer to A, a contradiction. If the value of O_i is either x or $x + 1/2$, O_i contains at least $\lfloor s \rfloor$ light goods. We augment the path to O and move $\lfloor s \rfloor$ light goods from i to j. This improves the NSW of O if the value of O_i is x and does not change the utility profile of O and moves O closer to A, otherwise. If the value of O_i is $x+1$, O_i contains at least $\lceil s \rceil$ light goods. We augment the path to O, move $\lfloor s \rfloor$ light goods from i to j and one light good from i to a bundle of value x. This improves O. In either case, we have obtained a contradiction.

If all bundles in A_1 are heavy-only, $A_1 \subseteq O_1$ and hence A is optimal by Lemma 13. Q.E.D.

In the rest of this section, we briefly summarize what we have obtained so far. Let A be reduced and shrunken and let O be optimal and closest to A. Assume further that Lemma 9g) is not applicable to A. The minimum value of any bundle in A is x and k_0 is minimal such that $k_0 s > x+1$.

- The bundles in A have value x, $x+1/2$, or $x+1$, and there is a bundle of value x.
- $x-1/2 \le x_O \le x+1/2$.
- In A and O, bundles contain at most k_0-1 heavy goods. Any bundle of value more than $(k_0-1)s$ must contain a light good.
- If A_i has value $x+1$ and is heavy-only, O_i has value $x+1$ and is heavy-only. If all bundles of value $x+1$ in A are heavy-only, A is optimal. Conversely, if A is suboptimal, there is a bundle of value $x+1$ in A containing a light good.
- Bundles in O have value at least x_O and at most $x+1$. Since $x_O \ge x-1/2$, bundles in O have values in $\{x-1/2, x, x+1/2, x+1\}$.

In the next section, we will introduce improving walks as an additional improvement rule and then show that an allocation to which no improvement is applicable is optimal. For the optimality proof, we consider a suboptimal allocation and an optimal allocation closest to it and then exhibit an applicable improvement rule. In the light of Lemma 16, we may assume that A contains a bundle of value $x+1$ containing a light good. The goal of the improvement rules is to create more bundles of value $x+1/2$.

Lemma 17. *Let A and A' be allocations in which all bundles have value in $\{x, x+1/2, x+1\}$, the total value of the bundles is the same, and A' contains more bundles of value $x+1/2$. Then the NSW of A' is higher than the NSW of A.*

Proof. For $d \in \{x, x+1/2, x+1\}$, let a_d and a'_d be the number of bundles of value $x+d$ in A and A', respectively. From $\sum_d a_d = \sum_d a'_d$, we conclude $a'_{1/2} - a_{\frac{1}{2}} = a_1 + a_0 - a'_1 - a'_0$. From $\sum_d d a_d = \sum_d d a'_d$, we conclude $a'_{1/2} - a_{\frac{1}{2}} = 2(a_1 - a'_1)$ and further $a_1 - a'_1 = a_0 - a'_0$. Let $z = a_1 - a'_1$. Then

$$\frac{\mathrm{NSW}(A')}{\mathrm{NSW}(A)} = \frac{(x+1/2)^{2z}}{x^z(x+1)^z} = \left(\frac{x^2+x+1/4}{x^2+x}\right)^z > 1.$$

Q.E.D.

Improving Walks. As already mentioned in the introductory section on improvement rules (Section 4.1), we need more general improving structures than alternating paths. We need improving walks which we introduce in this section. Improving walks are also used in the theory of parity matchings, i.e., generalized matchings in which degrees are constrained to a certain parity; see, for example, the chapter on parity factors in Akiyama et al. [1].

Let A be reduced and let O be optimal and closest to A. Our goal is to show that, whenever A is suboptimal, an improving walk exists. Improving walks use only edges in $A^H \oplus O^H$. As a first step, we show that for a suboptimal A there is an agent $i \in (A_0 \cup A_1) \cap O_{1/2}$ and that for such an agent $|A_i^H| \ne |O_i^H|$.

Lemma 18. *Let A be reduced and shrunken, let x be the minimum value of any bundle in A, and let O be any allocation in which all bundles have value at least $x - 1/2$ and at most $x+1$. For $d \in \{-1/2, 0, 1/2, 1\}$, let a_d and o_d be the number of bundles of value $x+d$ in A and O respectively, and let $z = a_1 - o_1$.*

a) *The parity of the number of heavy goods is the same in bundles of value x and $x+1$ and in bundles of value $x - 1/2$ and $x + 1/2$ and the former parity is different from the latter.*

b) *The parity of the number of bundles of value x or $x+1$ is the same in A and O and equally for the number of bundles of value $x - 1/2$ or $x + 1/2$. More precisely, (the first equation is trivial; it is there for completeness)*

$$a_{-1/2} = 0 = o_{-1/2} - o_{-1/2} \qquad a_0 = o_0 + 2o_{-1/2} + z$$
$$a_{1/2} = o_{1/2} - 2z - o_{-1/2} \qquad a_0 + a_1 = o_0 + o_1 + 2(o_{-1/2} + z).$$

c) *Let A be a suboptimal allocation and let O be an optimal allocation. Then $z > o_{-1/2} \geq 0$, $a_0 > 0$, $a_1 > 0$, and $(A_0 \cup A_1) \cap O_{1/2}$ is non-empty. In particular, O contains a bundle of value $x + 1/2$.*

Proof. If the values of two bundles differ by an integral amount, the numbers of heavy goods in both bundles differ by an even number. If the values differ by a multiple of $1/2$ which is not an integer, the numbers of heavy goods differ by an odd integer. This proves the first claim.

For the second claim, observe that $\sum_d a_d = \sum_d o_d$, $\sum_d a_d d = \sum_d o_d d$, and $a_{-1/2} = 0$. Thus

$$a_{1/2} = 2(\sum_d o_d d - a_1) = o_{1/2} - o_{-1/2} + 2(o_1 - a_1) = o_{1/2} - 2z - o_{-1/2},$$
$$a_0 + a_1 = \sum_d o_d - a_{1/2} = o_{-1/2} + o_0 + o_{1/2} + o_1 - a_{1/2} = o_0 + o_1 + 2(o_{-1/2} + z),$$

and finally

$$a_0 = o_0 + o_1 + 2(o_{-1/2} + z) - a_1 = o_0 + 2o_{-1/2} + z.$$

So $a_0 + a_1$ and $o_0 + o_1$ have the same parity. As a consequence, $a_{-1/2} + a_{1/2}$ and $o_{-1/2} + o_{1/2}$ also have the same parity. This proves the second claim.

We come to the third claim.

If $x = 0$, $o_{-1/2} = 0$. Since A is suboptimal, the NSW of O is positive and hence $o_0 = 0$. Also, $z > 0$ by Lemma 13. Thus $a_0 > o_0$, $a_1 > o_1$, $a_0 + a_1 > o_0 + o_1$ and hence $(A_0 \cup A_1) \cap O_{1/2} \neq \emptyset$.

If $x > 0$, we have

$$1 < \frac{\text{NSW}(O)}{\text{NSW}(A)} = \frac{(x - 1/2)^{o_{-1/2}} (x + 1/2)^{2z} (x + 1/2)^{o_{-1/2}}}{x^{2o_{-1/2}} x^z (x+1)^z} = \left(\frac{x^2 - 1/4}{x^2}\right)^{o_{-1/2}} \cdot \left(\frac{x^2 + x + 1/4}{x^2 + x}\right)^z,$$

and hence $z > o_{-1/2}$ since $(x^2 - 1/4)(x^2 + x + 1/4)/(x^2(x^2 + x)) < 1$. Thus

$$a_0 + a_1 = o_0 + o_1 + 2(o_{-1/2} + z) > o_0 + o_1 + o_{-1/2}$$

and hence $(A_0 \cup A_1) \cap O_{1/2} \neq \emptyset$ and $o_{1/2} = a_{1/2} + 2z + o_{-1/2} > 0$. Also, $a_0 \geq z > 0$ and $a_1 = o_1 + z > o_1$.
Q.E.D.

Remark: It is not true that $A_0 \cap O_{\pm 1/2}$ is guaranteed to be non-empty. Same for $A_1 \cap O_{\pm 1/2}$.

We will next prove a number of lemmas that guarantee ownership of light goods for certain agents. The *heavy parity* of a bundle is the parity of the number of heavy goods in the bundle. A node v is *unbalanced* if $|A_v^H| \neq |O_v^H|$. A node v is *A-heavy* if $|A_v^H| > |O_v^H|$ and *O-heavy* if $|A_v^H| < |O_v^H|$.

LEMMA 19. *Let v be unbalanced.*

a) $v \in ((A_0 \cup A_1) \cap (O_0 \cup O_1)) \cup (A_{1/2} \cap (O_{-1/2} \cup O_{1/2})$*:*
- *If v is A-heavy, O_v contains at least $2s - 1$ light goods if $v \in O_0 \cap A_1$ or $v \in A_{1/2} \cap O_{-1/2}$, and at least $2s$ light goods otherwise.*
- *If v is O-heavy, A_v contains at least $2s - 1$ light goods if $v \in O_1 \cap A_0$ and at least $2s$ light goods otherwise.*

b) $v \in (A_0 \cup A_1) \cap O_{\pm 1/2}$*:*
- *If v is A-heavy, O_v contains at least $\lfloor s \rfloor$ light goods if $v \in O_{1/2}$ and at least $\lfloor s \rfloor - 1$ light goods if $v \in O_{-1/2}$.*
- *If v is O-heavy, A_v contains at least $\lfloor s \rfloor$ light goods. If $v \in A_1$, O_v contains at least $\lceil s \rceil$ light goods.*

c) $v \in A_{1/2} \cap (O_0 \cup O_1)$*:*
- *If v is A-heavy, O_v contains at least $\lfloor s \rfloor$ light goods if $v \in O_0$ and at least $\lceil s \rceil$ light goods if $v \in O_1$.*
- *If v is O-heavy, A_v contains at least $\lfloor s \rfloor$ light goods. If $v \in O_0$, A_v contains at least $\lceil s \rceil$ light goods.*

Proof.

a) A_v and O_v have the same heavy parity. If v is A-heavy, $|A_v^H| \geq |O_v^H| + 2$. Hence the number of light goods in O_v is at least $2s$ minus the value difference between A_v and O_v. This value difference is non-positive except if $v \in A_1 \cap O_0$ or $v \in O_{-1/2} \cap A_{1/2}$. In these cases, the value difference is 1. If v is O-heavy, a symmetric argument applies.

b) If v is A-heavy, $|O_v^H| < |A_v^H|$. If $v \in O_{1/2}$, the value of O_v is at most $1/2$ less than the value of A_v and hence O_v contains at least $\lfloor s \rfloor$ light goods. If $v \in O_{-1/2}$, the value of O_v is at most $3/2$ less than the value of A_v and hence O_v contains at least $\lfloor s \rfloor - 1$ light goods.

If v is O-heavy, $|A_v^H| < |O_v^H|$. If $v \in A_0$, the value of A_v is at most $1/2$ lower than the value of O_v and hence A_v contains at least $\lfloor s \rfloor$ light goods. If $v \in A_1$, A_1 contains at least $\lceil s \rceil$ light goods.

c) If v is A-heavy, $|A_v^H| \geq |O_v^H| + 1$. If $v \in O_0$, the value of O_v is $1/2$ lower than the value of Av and hence O_v contains at least $\lfloor s \rfloor$ light goods. If $v \in O_1$, O_v contains at least $\lceil s \rceil$ light goods.

If v is O-heavy, $|A_v^H| \leq |O_v^H| - 1$. If $v \in O_1$, A_v contains at least $\lfloor s \rfloor$ light goods, if $v \in O_0$, A_v contains at least $\lceil s \rceil$ light goods.

Q.E.D.

The type of an edge is either A or O and we use T for the generic type. If $T = A$, $\bar{T}$ denotes O. If $T = O$, $\bar{T}$ denotes A. An *A-O-walk* from i to j is a sequence of $i = h_0$, e_0, h_1, $\ldots e_{\ell-1}$, $h_{\ell-1}$, e_ℓ, $h_\ell = j$ of agents, goods, and edges such that:

a) i and j are agents, i is unbalanced and lies in $(A_0 \cup A_1) \cap O_{1/2}$, j is unbalanced and $j \in A_0 \cup A_1 \cup O_0 \cup O_1$ (i.e., $j \notin A_{1/2} \cap O_{\pm 1/2}$).

b) All edges belong to $A^H \oplus O^H$ and the edges of the walk are pairwise distinct.

c) For $1 \leq t < \ell$, h_t is called a *through-node* if the edges e_t and e_{t+1} have different types and a *T-hinge* if both edges have type T. Hinges lie in $A_{1/2} \cap O_{\pm 1/2}$ and are unbalanced.

For a good h_t, the edges e_{t-1} and e_t have different types (one in A^H, one in O^H). The nodes i and j are the endpoints of the walk and h_1 to $h_{\ell-1}$ are intermediate nodes. The type of i is the type of e_0 and the type of j is the type of e_ℓ. Goods have degree zero or two in $A^H \oplus O^H$. We will augment A-O-walks to either A or O. Augmentation to A will improve A and augmentation to O will either improve O or move O closer to A. We allow $i = j$; we will augment such A-O-walks to O. There is no requirement on ownership of light goods by hinges and endpoints. We will later show that A-hinges own $2s$ light goods in O and O-hinges own $2s$ light goods in A and that endpoints own an appropriate number of light goods.

Lemma 20. *Let W be an A-O-walk. Then $|W \cap A| = |W \cap O|$.*

Proof. Every good of the walk is adjacent to one A-edge and one O-edge, and the edges of a walk are pairwise distinct. Q.E.D.

Lemma 21. *If A is sub-optimal, a non-trivial A-O-walk exists. Let i and j be the endpoints of the walk. If $i = j$ and the walk starts and ends with an edge of the same type, $|A_i^H|$ and $|O_i^H|$ differ by at least two.*

Proof. We construct the walk as follows. The walk uses only edges in $D = A^H \oplus O^H$ and visits each good at most once. We start with a node $i \in (A_0 \cup A_1) \cap O_{1/2}$; by Lemma 18 such a node exists. For such a node the parities of $|A_i^H|$ and $|O_i^H|$ differ. If $|A_i^H| > |O_i^H|$, we start tracing a walk starting at i with an A-edge, otherwise, we start with an O-edge.

Suppose we reach a node h on a T-edge e where $T \in \{A, O\}$. If there is an unused edge, i.e., not part of the walk, of type $\bar{T}$ incident to h, we continue on this edge. This will always be the case for goods. We come back to this claim below.

So assume that there is no unused edge of type $\bar{T}$ incident to h. Then h is an unbalanced T-heavy agent. This can be seen as follows. Any visit to a node uses edges of different types for entering and leaving the node as long as an unused edge of a different type is available for leaving the node. Any later visit either uses the same type for entering and leaving or uses up the last unused edge incident to the node.

If $h \in A_0 \cup A_1 \cup O_0 \cup O_1$, we stop. Then set $j = h$. If, in addition, $h = i$, the first and the last edge of the walk have the same type, say T, and the number of T-edges incident to i is at least two more than the number of $\bar{T}$-edges.

Otherwise, $h \in A_{1/2} \cap O_{\pm 1/2}$ and hence the number of heavy edges of both types incident to h has the same parity. Thus there is an unused T-edge incident to h. We pick an unused T-edge incident to h and continue on it.

Since the walk always proceeds on an unused edge and the first visit to a good uses up the A- and the O-edge incident to it, the walk visits each good at most once. Q.E.D.

A walk may pass through the same agent several times. A walk is *semi-simple* if for different occurrences of the same agent, the incoming edges have different types. In particular, any agent can appear at most twice. Goods appear at most once in a walk. If a walk exists, a semi-simple walk exists. We will not use this fact in the sequel of the paper, but state it for completeness.

LEMMA 22. *If there is an A-O-walk, there is a semi-simple walk with the same endpoints.*

Proof. Consider the walk and assume that an agent v is entered twice on an edge of the same type, say once from g and once from g'; the second occurrence of v could be the last vertex of the walk.

$$\ldots \text{—}\, g \overset{T}{\text{—}} v \overset{T'}{\text{—}} \ldots \text{—}\, g' \overset{T}{\text{—}} v \overset{T''}{\text{—}} \ldots$$

If the second occurrence of v is the last vertex of the walk, we end the walk at the preceding occurrence of v. Otherwise, the edge of type T'' exists. Either occurrence of v could be a hinge. We cut out the subpath starting with the first occurrence of v and ending with the edge entering the second occurrence of v and obtain

$$\ldots \text{—}\, g \overset{T}{\text{—}} v \overset{T''}{\text{—}} \ldots$$

If $T'' \neq T$, we still have a walk. If $T'' = T$, the second occurrence of v is a hinge and hence $v \in A_{1/2} \cap O_{\pm 1/2}$. After the removal of the subpath, it is still a hinge. Q.E.D.

Hinge nodes lie in $A_{1/2} \cap O_{\pm 1/2}$. If none of the basic improvement rules applies to A, hinge nodes actually lie in $O_{1/2}$ and T-hinges own $2s$ light goods in the allocation $\bar{T}$ for $T \in \{A, O\}$ as we show next.

LEMMA 23. *If A is reduced, all hinge nodes belong to $O_{1/2}$, A-hinges own at least $2s$ light goods in O and O-hinges own at least $2s$ light goods in A.*

Proof. By definition, the hinge nodes are unbalanced and lie in $A_{1/2} \cap O_{\pm 1/2}$. Consider two consecutive hinges h and h' and the alternating path P connecting them. Assume that their values in O differ by one, i.e, one has value $x + 1/2$ and the other value $x - 1/2$. The A-endpoint of the path owns at least $2s - 1$ light goods in O according to Lemma 19. When we augment the path to O, the A-endpoint receives an additional heavy good and the O-endpoint loses a heavy good. Depending on whether the A-endpoint is the heavier endpoint or not, we move $\lceil s \rceil$ or $\lfloor s \rfloor$ light goods to the other endpoint. This improves the NSW of O, a contradiction. We have now shown that all hinge nodes have the same value in O.

It remains to show that the first hinge of the walk lies in $O_{1/2}$; call it h. Assume $h \in O_{-1/2}$. We distinguish cases according to whether i is A-heavy or not.

If i is O-heavy, h is A-heavy and hence owns at least $2s - 1$ light goods in O (Lemma 19). We augment P to O and move $\lfloor s \rfloor$ light goods from h to i. After the change i and h belong to O_0 and the NSW of O has improved, a contradiction.

If i is A-heavy, i owns at least $2s - 1$ light goods in O (Lemma 19). We augment P to O and move $\lceil s \rceil$ light goods from i to h. After the change i and h belong to O_0 and the NSW of O has improved, a contradiction.

So, all hinges are unbalanced and belong to $A_{1/2} \cap O_{1/2}$. Thus Lemma 19 applies and A-hinges own at least $2s$ light goods in O and O-hinges own at least $2s$ light goods in A. Q.E.D.

At this point, we have established the existence of an A-O-walk with endpoint $i \in (A_0 \cup A_1) \cap O_{1/2}$. If A is reduced, all hinge nodes belong to $A_{1/2} \cap O_{1/2}$ and T-hinges own $2s$ light goods in the allocation $\bar{T}$. We will next show that we can use the A-O walk to improve A. This might require the existence of a facilitator. We therefore apply Lemma 9g) if possible. We distinguish cases according to whether i is O-heavy or A-heavy.

Case** $i \in (A_0 \cup A_1) \cap O_{1/2}$ **and** i **is** O**-heavy. The value of A_i is x or $x + 1$ and the value of O_i is $x + 1/2$ and A_i contains fewer heavy goods than O_i. Therefore A_i contains at least $\lceil s \rceil$ light goods if $i \in A_1$ and at least $\lfloor s \rfloor$ light goods if $i \in A_0$ by Lemma 19. Let W be an A-O-walk starting in i and let j be the endpoint of the walk. The types of the hinges alternate along the path, the type of the first (last) hinge is opposite to the type of i (j). Each A-hinge holds $2s$ light goods in O and each O-hinge holds $2s$ light goods in A (Lemma 23). If the types of i and j differ, the number of hinges is even, if the types are the same, the number of hinges is odd. There is also a bundle of value $x+1$ in A containing a light good and there is a bundle of value x in A.

We distinguish three cases: $j \in A_0 \cup A_1$ and $i \neq j$, $j \in A_0 \cup A_1$ and $i = j$, and $j \in (O_0 \cup O_1) \cap A_{1/2}$. In the first case, we show how to improve A and in the other two cases we will derive a contradiction to the assumption that O is closest to A.

Case $j \in A_0 \cup A_1$***, and*** $i \neq j$***:*** We augment the walk to A. The heavy parity of i and j changes; i gains a heavy edge and j gains or loses one. The heavy parity of all intermediate nodes does not change. Each O-hinge releases $2s$ light goods and each A-hinge requires $2s$ light goods. If j is an O-endpoint, j gains a heavy edge and the number of A-hinges exceeds the number of O-hinges by one.

Assume first that j is an O-endpoint. Each endpoint gives up $\lceil s \rceil$ light goods if in A_1 and $\lfloor s \rfloor$ light goods if in A_0. Note that by Lemma 19, i and j own that many light goods. So i and j together give up between $2s-1$ and $2s+1$ light goods; $2s$ of them are needed for the extra A-hinge. If they give up $2s+1$ and hence $i, j \in A_1$, one goes to an arbitrary bundle in A_0, and if they give up $2s-1$ and hence $i, j \in A_0$, we take a light good from an arbitrary bundle in A_1 owning a light good. This bundle plays the role of a facilitator for the transformation. Recall that if A is suboptimal, there is a bundle in A_1 containing a light good.

If j is an A-endpoint, i gains a heavy edge and j loses a heavy edge, i has $\lfloor s \rfloor$ or $\lceil s \rceil$ light goods. We move $\lfloor s \rfloor$ light goods from i to j. If $j \in A_0$, we put an additional light good on j which we take from a bundle in A_1 containing a light good, otherwise. The bundle in A_1 plays the role of a facilitator for the transformation. If $j \in A_1$ and i had $\lceil s \rceil$ light goods, we put the extra light good on any bundle in A_0.

As a result of the transformation, the values of i and j become $x + 1/2$. Also, if $i, j \in A_1$, the value of some agent changes from x to $x+1$, and if $i, j \in A_0$, the value of an agent changes from $x+1$ to x. The value of no other agent changes. In either case, we did not change the total value of the bundles and increased the number of agents in $A_{1/2}$ by two and hence improved A. This is by Lemma 17.

Case $j \in A_0 \cup A_1$***, and*** $i = j$***:*** We augment the walk to O. For the intermediate nodes the heavy parity does not change. For i the heavy parity also does not change; it either gains and loses a heavy good or it loses two heavy goods. It remains to show that there are sufficiently many light goods to keep the values of all bundles in O unchanged.

If i loses and gains a heavy good, the number of A- and O-hinges is the same and we use the light goods released by the A-hinges for the O-hinges. If i loses two heavy goods, there is one more A-hinge and we use the light goods from the extra A-hinge for i. The change brings O closer to A, a contradiction to the choice of O. Hence this case cannot arise.

Case $j \in (O_0 \cup O_1) \cap A_{1/2}$***:*** We augment the walk to O. The heavy parity of the intermediate nodes does not change. The heavy parity of i and j changes; i loses a heavy good and j either gains or loses a heavy good. It remains to show that there are sufficiently many light goods to keep the utility profile of O unchanged.

The A- and O-hinges on the walk alternate and their numbers are either the same, if i and j have different types, or there is an extra A-hinge if i and j are both O-endpoints. Each A-hinge releases $2s$ light goods and each O-hinge requires $2s$ light goods.

If j is an O-endpoint, we use the $2s$ light goods provided by the extra A-hinge as follows: If $j \in O_0$, we give $\lceil s \rceil$ light goods to j and $\lfloor s \rfloor$ light goods to i, moving j to $O_{1/2}$ and i to O_0 and if $j \in O_1$, we give $\lfloor s \rfloor$ light goods to j and $\lceil s \rceil$ light goods to i, moving j to $O_{1/2}$ and i to O_1.

If j is an A-endpoint, it gains a heavy good. By Lemma 19, j owns $\lfloor s \rfloor$ light goods if $j \in O_0$ and owns $\lceil s \rceil$ light goods if $j \in O_1$. We move these goods to i. In either case, j moves to $O_{1/2}$ and i moves to $O_0 \cup O_1$.

In all cases, the utility profile of O does not change and O moves closer to A, a contradiction to our choice of O. So this case cannot arise.

Case $i \in (A_0 \cup A_1) \cap O_{1/2}$ ***and*** i ***is*** A***-heavy.*** The value of O_i is $x + 1/2$, the value of A_i is x or $x+1$, and A_i contains at least one more heavy good than O_i. We observe first that O_i contains at least $\lceil s \rceil$ light goods. If $i \in A_0$, the heavy value of O_i is at most $x - s$ and hence O_i contains at least $\lceil s \rceil$ light goods. If the value of A_i is $x+1$, A_i cannot be heavy-only since then O_i would also have value $x+1$ according to Lemma 16 (recall that we apply Lemma 9.g if possible)and hence the heavy value of O_i is at most $x+1-1-s$. So O_i contains at least $\lceil s \rceil$ light good by Lemma 19. Let W be an A-O-walk starting in i and let j be the other endpoint of the walk. The types of the hinges along the walk alternate, O-hinges hold $2s$ light goods in A, and O-hinges hold $2s$ light goods in O. If the types of i and j differ, there is an equal number of hinges of both types, if i and j are A-endpoints, there is an extra O-hinge on the walk. There is also a bundle of value $x+1$ in A containing a light good and there is a bundle of value x in A.

Similar to the case where i is O-heavy, we distinguish three cases: $j \in A_0 \cup A_1$ and $i \neq j$, $j \in A_0 \cup A_1$ and $i = j$, and $j \in (O_0 \cup O_1) \cap A_{1/2}$. In the first case, we show how to improve A and in the other two cases we will derive a contradiction to the assumption that O is closest to A.

Case $j \in A_0 \cup A_1$***, and*** $i \neq j$***:*** We augment the walk to A. The heavy parity of i and j changes and the heavy parity of all intermediate nodes does not change.

If i and j are A-endpoints, both lose a heavy edge and there is an extra O-hinge releasing $2s$ light goods. We give $\lfloor s \rfloor$ light goods to any endpoint in A_1 and $\lceil s \rceil$ light goods to any endpoint in A_0. So we need between $2s-1$ and $2s+1$ light goods. If we need only $2s-1$, we put the extra light good onto any bundle in A_0, if we need $2s+1$, we take one light good from any bundle in A_1 with a light good.

If j is an O-endpoint, j gains a heavy good. By Lemma 19, A_j contains $\lfloor s \rfloor$ light goods if $j \in A_0$ and contains $\lceil s \rceil$ light goods if $j \in A_1$. We give $\lfloor s \rfloor$ light goods to i if $i \in A_1$ and $\lceil s \rceil$ light goods if $i \in A_0$. If an extra light good is needed, we take it from a bundle in A_1, if we have one more light good than necessary, we put it on a bundle in A_0.

In either case, we increased the number of agents in $A_{1/2}$ by two and hence improved A.

Case $j \in A_0 \cup A_1$***, and*** $i = j$***:*** We augment the walk to O. The heavy parity of the intermediate nodes does not change and the heavy parity of i changes neither. It either gains and loses a heavy good and then the number of hinges is even or it gains two heavy goods and then the number of hinges is odd and there is an extra O-hinge. We show that there are sufficiently many light goods to keep the values of all bundles in O unchanged.

This is obvious, if i loses and gains a heavy good. Then there is an equal number of hinges of both types and we simply move the light goods between them.

If i gains two heavy goods, the first and the last edge of the walk are A-edges. Hence $|A_i^H| \geq |O_i^H| + 2$ (Lemma 21). Since $i \in O_{1/2}$, the parity of the number of heavy goods in A_i and O_i is different. Thus, $|A_i^H| \geq |O_i^H| + 3$. Since the value of O_i is by at most $1/2$ lower than the value of A_i, O_i contains at least $2s$ light goods. We give $2s$ light goods to the extra O-hinge. Note that i gains two heavy goods and hence the value of O_i does not change.

We have now moved O closer to A, a contradiction to our choice of O. Thus this case cannot arise.

Case $j \in (O_0 \cup O_1) \cap A_{1/2}$***:*** We augment the walk to O. For the intermediate nodes, the heavy parity does not change. For i and j the heavy parity changes; i gains a heavy good and j either loses a heavy good and then the number of hinges is even or gains a heavy good and then the number of hinges is odd and there is an extra O-hinge. There are sufficiently many light goods to keep the values of all bundles in O unchanged, except for the bundles of i and j; i and j change values. Recall that O_i contains at least $\lceil s \rceil$ light goods.

If j is A-heavy, i and j gain a heavy good and there is an extra O-hinge. Since j is A-heavy, the heavy value of j in O is at most $x + 1 - s$. Thus j owns at least $\lceil s \rceil$ light goods if $j \in O_1$ and at least $\lfloor s \rfloor$ light goods if $j \in O_0$. If $j \in O_1$, we move $\lfloor s \rfloor$ light goods from i and $\lceil s \rceil$ light goods from j to the extra O-hinge. If $j \in O_0$, we move $\lceil s \rceil$ light goods from i and $\lfloor s \rfloor$ light goods from j to the extra O-hinge. In either case, the values of i and j interchange. Thus the utility profile of O does not change and O moves closer to A, a contradiction.

If j is O-heavy, j loses a heavy good and the number of hinges is even. If $j \in O_1$, we move $\lfloor s \rfloor$ light goods from i to j, if $j \in O_0$, we move $\lceil s \rceil$ light goods from i to j. In either case i and j swap values. Thus the utility profile of O does not change and O moves closer to A, a contradiction.

We have now established the existence of improving walks.

LEMMA 24. *If A is sub-optimal, an improving A-O-walk exists.*

4.3. The Algorithm Let A be reduced. Let x be the minimum value of any bundle and let k_0 be minimal such that $k_0 s > x + 1$. Lemma 11 tells us that an optimal allocation consists of all bundles of value at least $k_0 s$ in A plus an optimal allocation of $A_{low} = A_0 \cup A_{1/2} \cup A_1$. So our algorithm consists of three steps.

We start with an arbitrary allocation A and reduce it.

Let x and k_0 as above. We remove all bundles of value $k_0 s$ or more from A and are left with bundles of value x, $x + {}^1\!/_2$ and $x + 1$. The removed bundles also exist in an optimal allocation.

We then optimize A_{low} by repeated augmentation of improving walks by exploiting a connection to matchings with parity constraints as described next.

Matchings with Parity Constraints. Consider a generalized bipartite matching problem, where for each node v of a graph G we have a constraint concerning the degree of v in the matching M. We are interested in parity constraints of the form $\deg_M(v) \in \{p_v, p_v + 2, p_v + 4, \ldots, p_v + 2r_v\}$, where p_v and r_v are non-negative integers. Matchings with parity constraints can be reduced to standard matching, Tutte [37], Tutte [38], Lovasz [33], Cornuejols [20], Sebo [35]. For completeness, we review the construction given in Cornuejols [20].

Consider any node v and let t_v be the degree of v. We may assume $p_v + 2r_v \le t_v$; otherwise decrease r_v. We replace v by the following gadget. We have t_v vertices v_1 to v_{t_v} and $t_v - p_v$ vertices z_1 to $z_{t_v - p_v}$. We refer to them as v-vertices and z-vertices respectively. We connect each v_i with each z_j. Finally, we create the edges (z_1, z_2), …, $(z_{2r_v - 1}, z_{2r_v})$. Note that $t_v - p_v - 2r_v$ of the z-nodes are not incident to one of these inter-z-edges. This ends the description of the gadget for v. For every edge (v, w) of the original graph, we have the complete bipartite graph between the vertices v_i and w_j of the auxiliary graph.

LEMMA 25 (**[20]**). *The auxiliary graph has a perfect matching if and only if the original graph has a matching satisfying the parity constraints.*

Proof. We include a proof for completeness. Assume first that G has a matching M satisfying the parity constraints. We construct a perfect matching P in the auxiliary graph. For each edge $(v, w) \in M$, we pick one of the edges (v_i, w_j) in the auxiliary graph such that at most one edge incident to any v_i or w_j is picked. Then $p_v + 2\ell_v$ with $\ell_v \in \{0, 1, 2, \ldots, r_v\}$ of the v-vertices are matched outside the gadget and $t_v - p_v - 2\ell_v$ are not. We match $t_v - p_v - 2r_v$ of them to the z-nodes that are not incident to any of the inter-z-edges, $2(r_v - \ell_v)$ with the two vertices in $r_v - \ell_v$ pairs of z-nodes connected by an inter-z-edge and use inter-z-edges for the remaining pairs of z-nodes. In this way, we obtain a perfect matching in the auxiliary graph.

Conversely, assume that there is a perfect matching P in the auxiliary graph. Then P contains ℓ_v of the inter-z-edges for some $\ell_v \in \{0, 1, \ldots, r_v\}$ and hence $t_v - p_v - 2\ell_v$ of the z-vertices are matched with v-vertices. Thus $p_v + 2\ell_v$ v-vertices are matched outside the gadget. Remove the z-vertices and collapse all v-vertices into v. In this way, we obtain a matching M in G satisfying the parity constraints. Q.E.D.

As usual, let n and m be the number of edges in the input graph. The number of vertices of the auxiliary graph is $O(m)$ and the number of edges of the auxiliary graph is $O(\sum_v \deg_v^2 + \sum_{(v,w)\in E} \deg_v \deg_w) = O(mn^2)$ since degree in the original graph are bounded by n.

The Reduction to Parity Matching. Let g be the maximum number of heavy goods that a bundle of value $x + 1/2$ may contain. Then $g = \lfloor (x + 1/2)/s \rfloor$, The following lemma gives the maximum number of heavy goods in bundles of value x and $x + 1$.

Lemma 26. *Let g be the maximum number of heavy goods that a bundle of value $x + 1/2$ may contain. The following table shows the maximum number of heavy goods in bundles of value x and $x + 1$. We use $s\mathbb{N}$ to denote $\{s \cdot t \mid t \in \mathbb{N}\}$.*

x	$x + 1/2$	$x + 1$	*Condition*
$g - 1$	g	$g + 1$	$x + 1 \in s\mathbb{N}$
$g + 1$	g	$g + 1$	$x + 1 \notin s\mathbb{N}$ *and* $x + 1 > (g + 1)s$
$g - 1$	g	$g - 1$	$x + 1 \notin s\mathbb{N}$ *and* $x + 1 < (g + 1)s$

Clearly, $x + 1 \notin s\mathbb{N}$ implies $x + 1 \neq (g + 1)s$.

Proof. For $t \in \{0, 1/2, 1\}$, let m_t be the maximum number of heavy goods in a bundle of value $x + t$. Then $m_0 \le m_1 \le m_0 + 2$. The first inequality is obvious (add a light good to the lighter bundle) and the second inequality holds because we may remove two heavy goods from the heavy bundle and add $2s - 1$ light goods. Also m_0 and m_1 have the same parity. Finally $m_{1/2} - m_0 = \pm 1$ since the two numbers have different parity and we can switch between the two values by exchanging a heavy good by either $\lfloor s \rfloor$ or $\lceil s \rceil$ light goods.

Let $x + 1/2 = gs + y$ with $y \in \mathbb{N}_0$. Then $x + 1 = gs + y + 1/2$. If $y + 1/2 = s$, $m_1 = g + 1$. Also $m_0 < m_1$ and hence $m_0 = g - 1$. If $y + 1/2 > s$ then $y - 1/2 \ge s$ and therefore $m_1 = m_0 = g + 1$. If $y + 1/2 < s$, then $x + 1 = (g - 1)s + (s + y + 1/2)$ and $x - 1 = (g - 1)s + (s + y - 1/2)$ and therefore $m_1 = m_0 = g - 1$. Q.E.D.

We use N_0, $N_{1/2}$, and N_1 to denote the set of allowed number of heavy goods in bundles of value A_0, $A_{1/2}$ and A_1 respectively.

Lemma 27. *Let A be an allocation with all values in $\{x, x+1/2, x+1\}$. A is sub-optimal if and only if there is an allocation B^H of the heavy goods in A and a pair of agents i and j in $A_0 \cup A_1$ such that*

- *all agents in $A_{1/2} \cup \{i,j\}$ own a number of heavy goods in $N_{1/2}$ and for each of the agents in $A_0 \cup A_1 \setminus \{i,j\}$, the number of owned heavy goods is in the same N-set as in A, and,*
- *if i and j own bundles of value x in A, there must be a bundle of value $x+1$ in A containing a light good and if i and j own bundles of value $x+1$ in A, there must be a bundle of value x in A.*

Proof. If A is sub-optimal there is an improving walk W. Let i and j be the endpoints of the walk. Augmenting the walk and moving the light goods around as described in Section 4.2.3

- adds i and j to $A_{1/2}$,
- reduces the value of a bundle of value $x+1$ containing a light good to x if A_i and A_j have value x and increases the value of a bundle of value x to $x+1$ if A_i and A_j have value $x+1$, and
- leaves the value of all other bundles unchanged.

Thus, in the new allocation, the number of heavy goods owned by i and j lies in $N_{1/2}$. For all other agents the number of owned heavy goods stays in the same N-set. This proves the only-if direction.

We turn to the if-direction. Assume that there is an allocation B^H of the heavy goods in which for two additional agents i and j the number of owned heavy goods lies in $N_{1/2}$ and for all other agents the number of owned heavy goods stays in the same N-set. We will show how to allocate the light goods such that B^H becomes an allocation B, in which all bundles have value in $\{x, x+1/2, x+1\}$ and $B_{1/2} = A_{1/2} \cup \{i,j\}$. Then the NSW of B is higher than the one of A.

We next define the values of the bundles in B and in this way fix the number of light goods that are required for each bundle. For i and j, we define $v_i^B = v_j^B = x+1/2$. If A_i and A_j have both value x, let k be an agent owning a bundle of value $x+1$ containing a light good and define $v_k^B = x$. If A_i and A_j have both value $x+1$, let k be an agent owning a bundle of value x and define $v_k^B = x+1$. Then $v_i^A + v_j^A + v_k^A = v_i^B + v_j^B + v_k^B$ in both cases. If one of A_i and A_j has value x and the other one has value $x+1$, let k be undefined. Then $v_i^A + v_j^A = v_i^B + v_j^B$. For all ℓ different from i, j, and k, let $v_\ell^A = v_\ell^B$. Then the total value of the bundles in A and B is the same.

For an agent ℓ let h_ℓ and h'_ℓ be the number of heavy goods allocated to ℓ in A and B^H, respectively. Then $\sum_\ell h_\ell = \sum_\ell h'_\ell$. Moreover, $h_\ell \in N_{1/2}$ iff $\ell \in A_{1/2}$ and $h'_\ell \in N_{1/2}$ iff $\ell \in A_{1/2} \cup \{i,j\}$. For all $\ell \in A_0 \cup \{k\} \setminus \{i,j\}$, $h'_\ell \in N_0$ and for all $\ell \in A_1 \setminus \{i,j,k\}$, $h'_\ell \in N_1$. Then $v_\ell^B - sh'_\ell$ is a non-negative integer for all ℓ and

$$\sum_\ell (v_\ell^B - sh'_\ell) = \sum_\ell v_\ell^B - \sum_\ell sh'_\ell = \sum_\ell v_\ell^A - \sum_\ell sh_\ell = \sum_\ell (v_\ell^A - sh_\ell).$$

We conclude that the allocation B exists Q.E.D.

EXAMPLE 9. Let $s = 3/2$ and assume we have two agents owning bundles of value 3 and 4 respectively. We have either zero or two or four heavy goods and accordingly seven, four or one light good. Both agents like all heavy goods. We have $N_{1/2} = \{1\}$. If there are two heavy goods, the optimal allocation has two bundles of value $7/2$. If we have zero or four heavy goods, there is no way to assign exactly one heavy good to each agent and hence the optimal allocation has bundles of value 3 and 4. In the case of four heavy goods, 1 and 3 is possible. Both numbers are odd, but 3 is too large. ■

EXAMPLE 10. Let $s = 3/2$ and assume we have two agents owning bundles of value 2 and 3 respectively. The bundle of value 3 consists of two heavy goods and both agents like all heavy goods. We have $N_{1/2} = \{1\}$. Since the two agents have values x and $x+1$, there is no need for an agent k. In the optimal allocation both bundles contain a heavy and a light good. ■

In order to check for the existence of the allocation B^H, we set up the following parity matching problem for every pair i and j of agents.

- For goods the degree in the matching must be equal to 1.
- For all agents in $A_{1/2} \cup \{i, j\}$, the degree must be in $N_{1/2}$.
- If A_i and A_j have value x, let A_k be any bundle of value $x+1$ containing a light good. If A_i and A_j have value $x+1$, let A_k be a bundle of value x. The degree of k must be in N_0.[3]
- For an $a \in A_0 \setminus \{i, j, k\}$, the degree must be in N_0, and for an $a \in A_1 \setminus \{i, j, k\}$, the degree must be in N_1.

If B^H exists for some pair i and j, we improve the allocation. If B^H does notexist for any pair i and j, A is optimal.

Each improvement increases the size of $A_{1/2}$ by two and hence there can be at most $n/2$ improvements. In order to check for an improvement, we need to solve n^2 perfect matching problems in an auxiliary graph with m vertices and mn^2 edges.

LEMMA 28. *An optimal allocation for A_{low} can be computed in time $O(n^4 m^{3/2})$.*

Proof. A perfect matching in a graph with m vertices and n^2m edges can be constructed in time $O(m^{1/2}n^2m)$. In order to check for an improvement, one needs to solve n^2 matching problems, and there can be at most $n/2$ improvements. The time bound follows. Q.E.D.

We conjecture that this time bound can be improved.

[3] It would be incorrect to require that the degree of k must be in N_1 because we want to allocate a light good to k.

4.4. The General Case: Heavy Goods can be Allocated as Light Finally, we show our main theorem.

THEOREM 2. *There exists a polynomial-time algorithm computing a maximum NSW allocation for half-integral instances, i.e., when $q=2$ and p is an odd integer greater than two.*

Let O^t be a best allocation in which t heavy goods are allocated to agents that consider them light and let A^t be a best allocation in which t heavy goods are converted to light, i.e., the valuation functions are changed such that these t goods are light for all agents. We are interested in $\max_t \text{NSW}(O^t)$, but we will compute $\max_t \text{NSW}(A^t)$. Note that O^t might not exist. For example, if all goods are heavy for all agents then O^t does not exist for $t \geq 1$. If O^t does not exist, we abuse notation and define $\text{NSW}(O^t) = -\infty$.

LEMMA 29. $\max_t NSW(O^t) = \max_t NSW(A^t)$.

Proof. The allocation O^t is a contender for A^t and hence $\text{NSW}(O^t) \leq \text{NSW}(A^t)$ for all t. Let $t^* = \arg\max_t \text{NSW}(A^t)$ and assume that in A_{t^*} one of the converted heavy goods is allocated to an agent that considers it heavy. Re-converting the good to heavy improves the NSW, a contradiction to the choice of t^*. Q.E.D.

We follow the approach taken in the integral case in Section 3. We determine allocations A^t for $t = 0, 1, 2, \ldots$; we determined A^0 in the previous section (we used the notation A_0 there). In A^t, t goods are converted. Which t? We will next derive properties of the optimal set of converted goods.

For a set G of goods, let $C(G)$ be the allocation A^0 interpreted with the following modified valuation function: The goods in G are light for all agents and for any agent i, i owns the heavy goods $A_i^H \setminus G$ and $|A_i^H \cap G|$ light goods in addition to the light goods already owned by it.[4] Let $B(G)$ be an optimal allocation for the modified valuation function closest to $C(G)$, i.e., with minimal $|C^H(G) \oplus B^H(G)|$. We choose G such that

- $\text{NSW}(B(G))$ is maximum and
- among the sets G that maximize $\text{NSW}(B(G))$, $|G|$ is minimum.
- among the minimum cardinality sets G that maximize $\text{NSW}(B(G))$, $|C^H(G) \oplus B^H(G)|)$ is smallest.

For simplicity, let us write B and C instead of $B(G)$ and $C(G)$ for this choice of G. We also write x_j for the value of j's bundle in C and x'_j for the value of j's bundle in B. We use S to denote the set of agents i with $A_i^H \cap G \neq \emptyset$. In Lemmas 30, 31, and 32 we derive properties of G.

LEMMA 30. *Let G and S be as defined above and let x be the minimum value of any bundle in A^0. Then $x'_\ell \leq s + \min_{i \in S} x'_i \leq \min_{i \in S} x_i$ for all agents ℓ. For $i \in S$: B_i contains no light good, $B_i^H \subseteq A_i^H \setminus G$, $x - 1 \leq x'_i \leq x_i - |A_i^H \cap G| \leq x_i - s$, and $x_i > x + 1$.*

[4] We write A_i instead of A_i^0 for the bundles of A^0.

Proof. If G is empty, the Lemma obviously holds. So assume $G \neq \emptyset$ and hence $S \neq \emptyset$. Let $j = \arg\min_{i \in S} x'_i$ and assume there is an ℓ such that $x'_\ell > x'_j + s$. Then ℓ's bundle in B is heavy-only and hence ℓ owns at least two heavy items more than j in B. We return one of the converted items to j, reconvert it to a heavy, and replace it by a converted from ℓ. This changes x'_j to $x'_j + s$ and x'_ℓ to $x'_\ell - s$ and hence improves NSW by Lemma 1a).

For the other claims consider any $i \in S$ and let $g \in A_i^H \cap G$. Let j be the agent to which g is allocated in B; $j = i$ is possible.

Assume first that B_i contains a light good, say g'. We interchange g and g', i.e., we allocate g' to j and g to i. This does not change the NSW. We now reconvert g back to a heavy good and improve the NSW, a contradiction to the optimality of B. So, B_i contains no light good.

Assume next that $B_i^H \setminus C_i^H$ is non-empty. With respect to $B^H \oplus C^H$ goods have degree zero or two. Also $B^H \oplus C^H$ contains no cycles as augmenting a cycle to B^H would decrease the distance between C^H and B^H. Thus $B^H \oplus C^H$ is a collection of B-C-alternating paths. One of these paths, call it P, starts in i with a B-edge and ends with a C-edge (g', h) at some agent h. We augment P to B, re-allocate g to i as a heavy good, convert g' to a light good and give it to j. The values of all bundles stay unchanged. We obtain an allocation D with the same NSW as B, $G' = (G \setminus g) \cup g'$ as the set of converted goods, and $|D^H \oplus C^H(G')| < |B^H \oplus C^H|$, a contradiction to the choice of G. So, $B_i^H \subseteq C_i^H = A_i^H \setminus G$ and hence $x'_i \le x_i - |A_i^H \cap G| \le x_i - s$ since B_i contains no light good.

From the preceding and the first paragraph of the proof, we obtain $x'_\ell \le s + \min_{i \in S} x'_i \le s + \min_{i \in S} x_i - s = \min_{i \in S} x_i$ for any agent ℓ.

We next prove $x'_i \ge x - 1$ for all $i \in S$. If S comprises all agents, $x'_i \le x_i - s$ for all agents and hence B is not optimal. So assume that S does not comprise all agents and there is an agent $i \in S$ with $x'_i \le x - 3/2$. Consider the following allocation D. Starting with $C(G)$, we move for all $i \in S$, the light goods in C_i to agents outside S and the heavy goods in $C_i^H \setminus B_i^H$ to their owners in B. At this point, the allocation agrees with B for all $i \in S$. We then apply the optimization rules to the bundles outside S. We obtain an allocation with the same NSW as B. Since none of the optimization rules decreases the value of the minimum bundle, all bundles in D outside S have value at least x and one of them contains a light good. Moving this light good to i improves the NSW according to Lemma 9a), a contradiction to the optimality of B.

We finally show $x_i > x + 1$ for all $i \in S$. For $s \ge 5/2$, we have $x_i \ge x'_i + s \ge x - 1 + s > x + 1$. So assume $s = 3/2$: If $x_i = x$, we have $x'_i \le x - 3/2$, and if i's bundle in A_0 contains a light item, we have $x'_i \le x + 1 - 1 - s \le x - 3/2$, a contradiction to $x'_i \ge x - 1$. We are left with the i such that $x_i \in \{x + 1/2, x + 1\}$ and i's bundle in A_0 is heavy-only. Let $S' \subseteq S$ be the set of such i and assume $S' \neq \emptyset$. Let $i_0 \in S$ and let $g \in A_{i_0} \cap G$. For $i \in S'$, A_i contains exactly one item in G and $B_i^H = C_i^H$ as otherwise $x'_i < x - 1$, and the conversion of the items in G changes the value of i's bundle to a value of at most $x_i - s$ (which is less than x) and creates a light good. We cannot have that S' comprises all agents and bundles B_i, $i \in S'$, are heavy-only and there are light goods. We move any light item in a bundle in S' to a bundle outside S'. We now have $C_i = B_i$ for $i \in S'$. We next re-optimize the bundles outside S' to obtain B. Since the NSW of C is worse than the NSW of A_0, after re-optmization there will be an agent h owning a bundle of value $x + 1/2$ containing a light good. Note that bundles of value $x+1$ or more cannot contain a light good, because i's bundle has value less than x. Let j be the agent to which g is allocated. We move a light good from h to j and we move g from j to i_0 and reconvert it to a heavy good. The multiplicative change in NSW is at least

$$\frac{x - 1/2}{x + 1/2} \cdot \frac{x+1}{x - 1/2} = \frac{x+1}{x + 1/2}$$

and hence the change improves NSW, a contradiction. Q.E.D.

Corollary 5. *Let x be the minimum value of any bundle in A^0. If A^0 contains no bundle of value more than $x+1$, A^0 is optimal.*

Proof. If A^0 is not-optimal, $S \neq \emptyset$. Finally, $x_i > x+1$ for $i \in S$ by Lemma 30. Q.E.D.

Recall that $k_0 = \arg\min_k ks > x+1$ and that for $k \geq k_0$, R_k denotes the bundles with exactly k heavy items in A^0 and R'_k denotes the bundles with exactly $k-1$ heavy items in A^0 to which an heavy item can be pushed from R_k.

Lemma 31. *Let k_1 be such that $(k_1 - 1)s = \min_{i \in S} x'_i$. Then $k_1 \geq k_0$ and $R_{\geq k_1+1} \subseteq S \subseteq R_{\geq k_1}$. Also $B_i = C_i^H \setminus G$ for $i \in S$.*

Proof. Since no heavy item in a bundle of value $x+1$ or less is converted, we have $k_1 \geq k_0$. Also, $x'_i \in \{(k_1-1)s, k_1 s\}$ for all $i \in S$. So $S \subseteq R_{\geq k_1}$. Then G contains $\ell - k_1$ heavy items in any bundle in R_ℓ for $\ell > k_1$ and the remaining $r = |G| - \sum_{\ell > k_1} (\ell - k_1)|R_\ell|$ heavy items in $R_{\geq k_1}$. After the conversion, all bundles in $R_{\geq k_1}$ contain at most k_1 heavy items and at least r of them contain at most $k_1 - 1$ heavy items.

We next show $B_i^H = C_i^H$ for $i \in S$. Assume, for the sake of a contradiction, that there is an $i \in S$ with $C_i^H \setminus B_i^H \neq \emptyset$. Since $(k-1)s \leq x'_i < x_i - |A_i \cap G|s = ks$, we have $x'_i = (k-1)s$. As above, we conclude that $B^H \oplus C^H$ contains no cycle and hence decomposes into paths. Then there is an $B^H \oplus C^H$ alternating path P connecting i and h, starting with a C-edge in i and ending with a B-edge in h. Furthermore, the heavy degree of h in B is larger than its heavy degree in C. Since $B_h^H \not\subseteq C_h^H$, we have $h \notin S$ and hence $h \in R_{\leq k_1}$. Write $R_{\leq k_1} = (R_{<k_1} \setminus R'_{k_1}) \cup R'_{k_1} \cup R_{k_1}$.

We cannot have $h \in R_{<k_1} \setminus R'_{k_1}$ as agents in $R_{<k_1} \setminus R'_{k_1}$ consider goods owned by agents in $R'_{k_1} \cup R_{\geq k_1}$ light. More precisely, trace P from i until an agent $h' \in R_{<k_1} \setminus R'_{k_1}$ is reached. The good g' preceding h' is owned by an agent in $R'_{\geq k_1} \cup R_{\geq k_1}$ in C and hence considered light by h'.

We cannot have $h \in R_{k_1}$, because then $x'_h \geq (k_1+1)s$, a contradiction.

So assume $h \in R'_{k_1}$. Then $x'_h \geq ks$. Consider the allocation D obtained by augmenting P to B. The heavy degree of i increases to k and the heavy degree of h decreases to at most $k-1$. Thus the NSW does not decrease and D is closer to C than B, a contradiction to the choice of B.

Since B_i is heavy-only, $B_i^H = C_i^H$ implies $B_i = C_i^H$. Q.E.D.

LEMMA 32. *Either $G = \emptyset$ or for every $i \in N$ such that $v_i(A_i)$ is maximal and for every $g \in A_i$ there exists an optimal G such that $g \in G$.*

Proof. We may assume $G \neq \emptyset$. Let $i \in N$ be such that $v_i(A_i)$ is maximal and $g \in A_i$. We may assume $g \notin G$.

Assume first that $x'_i < x_i$ and i is incident to a C-edge in $C^H \oplus B^H$. Let P be a maximal alternating path starting at i with an edge in $C^H \setminus B^H$. Since every good has even degree in $C^H \oplus B^H$, P ends at an agent k with a B-edge. Then $k \notin S$ by Lemma 30 and, by Claim 5 from Section 4.2.2, A_k contains at most one heavy good less than A_i, so $x'_k \geq x_i > x'_i$. We augment P to B. If $x'_k \geq x'_i + s$, the NSW of B does not decrease. Otherwise, i has $\lceil x'_i - (x_i - s) \rceil$ light goods that we give to k. Now, i has value $x'_i + s - \lceil s + x'_i - x_i \rceil = x'_i + (x_i - x'_i - \delta)$ and k has value $x'_k - s + \lceil s + x'_i - x_i \rceil = x'_k - (x_i - x'_i - \delta)$, where $\delta \in \{0, 1/2\}$. Since the sum of their utilities does not change and $0 \leq x_i - x'_i - \delta \leq x'_k - x'_i$, the NSW of B does not decrease. But B moves closer to C, a contradiction to the choice of B. So we have either $x'_i \geq x_i$ or $C^H \subseteq B^H$.

Assume that $i \notin S$. If $x'_i < x_i$, then i is incident to a C-edge in $C^H \oplus B^H$, a case we have already excluded. Hence, $x'_i \geq x_i$. Since $G \neq \emptyset$, there is a $j \in S$. We have $x'_i \geq x_i = v_i(A_i) \geq v_j(A_j) \geq x'_j + s$. By Lemma 9.a), B_i is heavy-only. Let $g' \in B_i$, $h \in C_j \cap G$ and j' the owner of h in B. We convert g' to a light good and give it to j' and give h back to j as a heavy good. This does not decrease the NSW of B, does not change the size of G, and yields an allocation at least as close to C as B, so we may assume that $i \in S$.

If $i \in S$, $x'_i < x_i$ and hence $C^H \subseteq B^H$. Thus $g \in B_i$, so we may exchange g with any good in $A_i \cap G$ and the result holds. Q.E.D.

It is now easy to complete the proof of Theorem 2. We start with A^0. If there is no bundle of value more than $x(A^0)+1$, A^0 is optimal. Otherwise, we select any bundle of maximal value and any good in this bundle, convert the good to a light good and re-optimize to obtain A^1. If there is no bundle of value more than $x(A^1)+1$, we stop. Otherwise, we select a bundle of maximal value and any good in this bundle, convert the good to a light good and re-optimize to obtain A^2. We continue in this way and then select the best allocation among A^0, A^1, ...

Polynomial Time: We construct iteratively allocations A^1, A^2, ... Each time we convert a heavy good to a light good and re-optimize. There are at most m conversions and each re-optimization takes polynomial time $O(n^4 m^{3/2})$ by Lemma 28. Thus the overall time is polynomial.

5. NP-Hardness when $q \geq 3$ In this section, we complement our positive results on polynomial-time NSW optimization. In particular, we show:

THEOREM 3. *It is* NP*-hard to compute an allocation with optimal NSW for 2-value instances, for any constant coprime integers* $p > q \geq 3$*.*

We provide a reduction from the NP-hard *q-Dimensional-Matching (q-DM)*. Given a graph G consisting of q disjoint vertex sets $V_1, \ldots, V_q$, each of size n, and a set $E \subseteq V_1 \times \ldots \times V_q$ of m, $m \geq n$, edges, decide whether there exists a perfect matching in G or not. Note that for $q = 3$ the problem is the well-known 3-DM and thus NP-hard. NP-hardness for $q > 3$ follows by simply copying the third set of vertices in the 3-DM instance $q - 3$ times, thereby also extending the edges to the new vertex sets.

Transformation: There is one good for each vertex of G, call them *vertex goods*. Additionally, there are $p(m-n)$ *dummy goods*. For each edge of G, there is one agent who values the q incident vertex goods p/q and all other goods 1.

LEMMA 33. *If* G *has a perfect matching, then there is an allocation* A *of the goods with* $NSW(A) = p$*. If* G *has no perfect matching, then for any allocation* A *of goods,* $NSW(A) < p$*.*

Proof. Suppose there exists a perfect matching in G. We allocate the goods as follows: Give each agent corresponding to a matching edge all q incident vertex goods. Now there are $m - n$ agents left. Give each of them p dummy goods. As each agent has utility p, the NSW of this allocation is p as well.

For the second claim, assume there is an allocation $A = (A_1, \ldots, A_m)$ of goods with $\text{NSW}(A) \geq p$. We show that in this case there is a perfect matching in G. First, observe that if we allocate each good to an agent with maximal value for it, we obtain an upper bound on the average utilitarian social welfare of A, i.e., $1/m \sum_i v_i(A_i) \leq 1/m(qn \cdot p/q + p(m-n)) = p$. Applying the AM-GM inequality gives us $\text{NSW}(A) = \left(\prod_i v_i(A_i)\right)^{1/m} \leq p$, and, furthermore, $\text{NSW}(A) = p$ iff $v_i(A_i) = p$ for all agents i. Hence each agent's utility is p in A and each vertex good is allocated to an incident agent. The next claim allows us to conclude that there are only two types of agents in A:

CLAIM 12. *If an agent* i *has valuation* $v_i(A_i) = p$*, then she either gets her* q *incident vertex goods or* p *other goods.*

Proof. Let $i, j \in \mathbb{N}_0$ be such that $p = i \cdot p/q + j$. Then $j \leq p$ since $i \geq 0$, and $(p - j)q = ip$. Since p and q are co-prime, p divides j. Thus either $j = 0$ and $i = q$ or $j = p$ and $i = 0$. Q.E.D.

By Claim 12, an agent either receives 0 or q of its vertex goods. As there are qn vertex goods, and each of them must be given to an incident agent, there must be n agents receiving their q incident vertex goods, which implies that there is a perfect matching in G.

Lemma 33 yields the proof of Theorem 3.

6. Conclusions We have delineated the border between tractable and intractable 2-value NSW instances. If goods have weights 1 and p/q with co-prime p and q, instances with $q=1$ and $q=2$ are tractable and for $q \geq 3$ the problem of maximizing the NSW is NP-complete. We suggest three directions for further research.

Our algorithms, in particular for the half-integral case, are fairly complex. Find simpler algorithms.

Find a succinct certificate of optimality in the spirit of McConnell et al. [34], i.e., can one in addition to the optimal allocation compute a succinct and easy-to-check certificate that witnesses the optimality of the allocation. Such certificates are available for many generalized matching problems, see for example the book by Akiyama et al. [1]. An early example is Tutte's certificate for the non-existence of a perfect matching, Tutte [37]. An undirected graph G has no perfect matching if and only if there is a subset U of the vertices such that $odd(G-U) > |U|$, where $odd(G-U)$ is the number of connected components of $G-U$ of odd cardinality. Of course, a perfect matching witnesses the existence of a perfect matching.

In Akrami et al. [2] a 1.0345 approximation algorithm is given and APX-hardness is shown for $q \geq 4$. APX-hardness for $q=3$ is shown in [23]. It is not known whether the approximation factor is best possible.

Acknowledgments. Martin Hoefer and Giovanna Varricchio were supported by DFG grant Ho 3831/5-1.

References

[1] Akiyama J, Kano M (2011) *Factors and Factorizations of Graphs*, volume 2031 of *Lecture Notes in Mathematics* (Springer).

[2] Akrami H, Chaudhury BR, Hoefer M, Mehlhorn K, Schmalhofer M, Shahkarami G, Varricchio G, Vermande Q, Wijland Ev (2022) Maximizing Nash social welfare in 2-value instances. *Proc. 36th Conf. Artif. Intell. (AAAI)* 36(5):4760–4767.

[3] Aleksandrov M, Aziz H, Gaspers S, Walsh T (2015) Online fair division: Analysing a food bank problem. *Proc. 24th Intl. Joint Conf. Artif. Intell. (IJCAI)*, 2540–2546.

[4] Amanatidis G, Birmpas G, Filos-Ratsikas A, Hollender A, Voudouris AA (2021) Maximum Nash welfare and other stories about EFX. *Theor. Comput. Sci.* 863:69–85.

[5] Anari N, Gharan SO, Saberi A, Singh M (2017) Nash social welfare, matrix permanent, and stable polynomials. *Proc. 8th Symp. Innovations in Theoret. Computer Science (ITCS).*

[6] Anari N, Mai T, Gharan SO, Vazirani VV (2018) Nash social welfare for indivisible items under separable, piecewise-linear concave utilities. *Proc. 29th Symp. Discrete Algorithms (SODA)*, 2274–2290.

[7] Babaioff M, Ezra T, Feige U (2021) Fair and truthful mechanisms for dichotomous valuations. *Proc. 35th Conf. Artif. Intell. (AAAI)*, 5119–5126.

[8] Barman S, Bhaskar U, Krishna A, Sundaram R (2020) Tight approximation algorithms for p-mean welfare under subadditive valuations. *Proc. 28th European Symp. Algorithms (ESA)*, 11:1–11:17.

[9] Barman S, Krishnamurthy SK, Vaish R (2018) Finding fair and efficient allocations. *Proc. 19th Conf. Economics and Computation (EC)*, 557–574.

[10] Barman S, Krishnamurthy SK, Vaish R (2018) Greedy algorithms for maximizing Nash social welfare. *Proc. 17th Conf. Auton. Agents and Multi-Agent Systems (AAMAS)*, 7–13.

[11] Bouveret S, Lemaître M (2016) Characterizing conflicts in fair division of indivisible goods using a scale of criteria. *Auton. Agents and Multi-Agent Syst.* 30(2):259–290.

[12] Brown A, Laddha A, Pittu MR, Singh M (2024) Approximation algorithms for the weighted nash social welfare via convex and non-convex programs. Woodruff DP, ed., *Proceedings of the 2024 ACM-SIAM Symposium on Discrete Algorithms, SODA 2024, Alexandria, VA, USA, January 7-10, 2024*, 1307–1327 (SIAM), URL http://dx.doi.org/10.1137/1.9781611977912.52.

[13] Caragiannis I, Gravin N, Huang X (2019) Envy-freeness up to any item with high Nash welfare: The virtue of donating items. *Proc. 20th Conf. Economics and Computation (EC)*, 527–545.

[14] Caragiannis I, Kurokawa D, Moulin H, Procaccia AD, Shah N, Wang J (2019) The unreasonable fairness of maximum Nash welfare. *ACM Trans. Econom. Comput.* 7(3):12:1–12:32.

[15] Chaudhury BR, Cheung YK, Garg J, Garg N, Hoefer M, Mehlhorn K (2018) On fair division for indivisible items. *Proc. 38th Conf. Foundations of Software Tech. and Theoret. Comp. Sci. (FSTTCS)*, 25:1–25:17.

[16] Chaudhury BR, Garg J, Mehta R (2021) Fair and efficient allocations under subadditive valuations. *Proc. 35th Conf. Artif. Intell. (AAAI)*, 5269–5276.

[17] Chaudhury BR, Kavitha T, Mehlhorn K, Sgouritsa A (2020) A little charity guarantees almost envy-freeness. *Proc. 31st Symp. Discrete Algorithms (SODA)*, 2658–2672.

[18] Cole R, Devanur N, Gkatzelis V, Jain K, Mai T, Vazirani V, Yazdanbod S (2017) Convex program duality, Fisher markets, and Nash social welfare. *Proc. 18th Conf. Economics and Computation (EC)*, 459–460.

[19] Cole R, Gkatzelis V (2018) Approximating the Nash social welfare with indivisible items. *SIAM J. Comput.* 47(3):1211–1236.

[20] Cornuéjols G (1988) General factors of graphs. *J. Comb. Theory, Ser. B* 45(2):185–198.

[21] Darmann A, Schauer J (2015) Maximizing Nash product social welfare in allocating indivisible goods. *Europ. J. Oper. Res.* 247(2):548–559.

[22] Dobzinski S, Li W, Rubinstein A, Vondrák J (2023) A constant factor approximation for nash social welfare with subadditive valuations. *CoRR* abs/2309.04656, URL http://dx.doi.org/10.48550/ARXIV.2309.04656.

[23] Fitzsimmons Z, Viswanathan V, Zick Y (2024) On the hardness of fair allocation under ternary valuations.

[24] Freeman R, Sikdar S, Vaish R, Xia L (2019) Equitable allocations of indivisible goods. *Proc. 28th Intl. Joint Conf. Artif. Intell. (IJCAI)*, 280–286.

[25] Garg J, Hoefer M, Mehlhorn K (2018) Approximating the Nash social welfare with budget-additive valuations. *Proc. 29th Symp. Discrete Algorithms (SODA)*, 2326–2340.

[26] Garg J, Husić E, Li W, Végh LA, Vondrák J (2022) Approximating Nash social welfare by matching and local search. *arXiv preprint arXiv:2211.03883* .

[27] Garg J, Husic E, Végh L (2021) Approximating Nash social welfare under Rado valuations. *Proc. 53rd Symp. Theory of Computing (STOC)*, 1412–1425.

[28] Garg J, Kulkarni P, Kulkarni R (2020) Approximating Nash social welfare under submodular valuations through (un)matchings. *Proc. 31st Symp. Discrete Algorithms (SODA)*, 2673–2687.

[29] Garg J, Murhekar A (2021) Computing fair and efficient allocations with few utility values. *Proc. 14th Symp. Algorithmic Game Theory (SAGT)*, 345–359.

[30] Halpern D, Procaccia AD, Psomas A, Shah N (2020) Fair division with binary valuations: One rule to rule them all. *Proc. 16th Conf. Web and Internet Economics (WINE)*, 370–383.

[31] Lee E (2017) APX-hardness of maximizing Nash social welfare with indivisible items. *Inf. Process. Lett.* 122:17–20.

[32] Li W, Vondrák J (2021) A constant-factor approximation algorithm for Nash social welfare with submodular valuations. *Proc. 62nd Symp. Foundations of Computer Science (FOCS)*, 25–36.

[33] Lóvasz L (1970) Subgraphs with prescribed valencies. *Journal of Combinatorial Theory* 8:391–416.

[34] McConnell R, Mehlhorn K, Näher S, Schweitzer P (2011) Certifying algorithms. *Computer Science Review* 5(2):119–161, ISSN 1574-0137.

[35] Sebö A (1993) General antifactors of graphs. *J. Comb. Theory, Ser. B* 174–184.

[36] Suksompong W, Teh N (2022) On maximum weighted Nash welfare for binary valuations. *Mathematical Social Sciences* 117:101–108.

[37] Tutte W (1952) The factors of graphs. *Canad. J. Math* 314–328.

[38] Tutte W (1954) A short proof of the factor theorem for finite graphs. *Canad. J. Math* 347–352.