Fair allocation and linear programming

Uriel Feige

June 14, 2022

Abstract

The purpose of this lecture is to illustrate the versatility of linear programming, and show one way of rounding fractional solutions of linear programs into integral solutions. This rounding procedure is referred to as *faithful implementation*, and we present an application of it in the context of fair allocation of indivisible goods. The manuscript *Best-of-Both-Worlds Fair-Share Allocations* by Babaioff, Ezra, and Feige [2021] contains the relevant material for this topic (and more). For convenience of the students in the reading course, we present here (almost verbatim) the portions of that manuscript that are most relevant, and expand on them in some places.

1 Introduction

We consider fair allocation of indivisible items to agents with additive valuations. An *instance* $\mathcal{I} = (v, \mathcal{M}, \mathcal{N})$ of the fair allocation problem consists of a set $\mathcal{M}$ of m indivisible items, a set $\mathcal{N}$ of n agents, and vector $v = (v_1, v_2, \ldots, v_n)$ of non-negative additive valuations, with the valuation of agent $i \in \mathcal{N}$ for set $S \subseteq \mathcal{M}$ being $v_i(S) = \sum_{j \in S} v_i(j)$, where $v_i(j)$ denotes the value of agent i for item $j \in \mathcal{M}$. We assume that the valuation functions of the agents are known to the social planer, and that there are no transfers (no money involved). We further assume that all agents have equal entitlement to the items. An allocation A is a collection of n disjoint bundles $A_1, \ldots, A_n$ (some of which might be empty), where $A_i \subseteq \mathcal{M}$ for every $i \in \mathcal{N}$. A randomized allocation is a distribution over deterministic allocations. We wish to design randomized allocations that enjoy certain fairness properties.

Before discussing some standard fairness properties, we briefly motivate the *best of both worlds* (BoBW) framework, that considers both ex-ante and ex-post properties of randomized allocations. Consider a simple allocation instance $\mathcal{I}_1$ with two agents and two equally valued items. Intuitively, any fair allocation in this case is an allocation that gives each agent one of the items. Giving both items to one of the agents and no item to the other agent is not considered fair. Consider now an instance $\mathcal{I}_2$ with two agents and just one item. As we want to allocate the item (to achieve Pareto efficiency) but the item is indivisible, we give it to one of the agents, and then the other agent gets no item. The fact that some agent receives no item is unavoidable, and in this respect the allocation is fair. Yet, the agent not getting the item might argue that this deterministic allocation is unfair as she has the same right to the item as the other agent. Indeed, we can improve the situation at least ex-ante: We can invoke a lottery to decide at random which of the two agents gets the item. While for any realization inevitably one agent is left with nothing, the allocation mechanism is ex-ante fair (each agent has a fair chance to win the lottery). Going back to instance $\mathcal{I}_1$, we could also have a lottery for $\mathcal{I}_1$, and have the winner receive both items. This too would be ex-ante fair, but ex-post (with respect to the final allocation) it would not be fair (as we did have the option to choose an allocation that gives every agent one item). Examples such as those above illustrate why we want our allocation mechanism to concurrently enjoy *both* ex-ante and ex-post fairness guarantees, as each guarantee by itself seems not to be sufficiently fair.

For the purpose of defining ex-ante fairness properties of randomized allocations, we assume that agents are risk neutral. That is, the ex-ante value that an agent derives from a distribution over bundles is the same as the expected value of a bundle selected at random from this distribution. Consequently, when considering a distribution D over allocations (of $\mathcal{M}$ to $\mathcal{N}$), we also consider the expectation of this distribution, which can be interpreted as a *fractional allocation*. In this fraction allocation, the fraction of item i given to agent j exactly equals the probability with which agent i receives item j under D. We naturally extend the additive valuation functions of agents to fractional allocations, by considering the expected valuation, that is, an additive valuation where the value of a fraction q_j of item j to agent i is $q_j \cdot v_i(j)$.

1.1 Notation and terminology

The *proportional share* of agent i is $PS_i = \frac{v_i(\mathcal{M})}{n}$. We say that an allocation $A = (A_1, \dots, A_n)$ is *proportional* if every agent i gets value at least PS_i (that is, $v_i(A_i) \geq \frac{v_i(\mathcal{M})}{n} = PS_i$), and a fractional (randomized) allocation is *ex-ante proportional* if she gets her proportional share in expectation. We say that an allocation A is *proportional up to one item (Prop1)* if for every agent i it holds that $v_i(A_i) \geq PS_i - \max_{j \in \mathcal{M} \setminus A_i}[v_i(j)]$.

An (fractional) allocation *Pareto dominates* another (fractional) allocation if it is weakly preferred by all agents, and strictly so by at least one agent. An integral allocation is *Pareto optimal (PO)* if no integral allocation Pareto dominates it. An allocation (integral or fractional) is *fractionally Pareto optimal (fPO)* if it is Pareto optimal, and moreover, no fractional allocation Pareto dominates it.

As we shall be dealing with randomized allocations, let us introduce terminology that we shall use in this context. A *random allocation* is a distribution D over integral allocations $A^1, A^2, \dots$. It induces an *expected* allocation A^*, where A^*_{ij} specifies for agent i and item j the probability that agent i receives item j, when an allocation is chosen at random from the underlying distribution

D. These probabilities can be interpreted as fractions of the item that an agent receives ex-ante. Hence the expected allocation A^* can be viewed as a *fractional allocation*, in which items are divisible. Conversely, we say that the distribution D (namely, the random allocation) *implements* the fractional allocation A^* when the expectation of D is A^*. Finally, we note that an additive valuation function can be extended in a natural way from allocations to fractional allocations, by considering the expected valuation. That is, the value of a p_j fraction of item j to agent i to is $p_j \cdot v_i(j)$, and the value of a fractional allocation A^* to agent i is $\sum_{j \in \mathcal{M}} A^*_{ij} \cdot v_i(j)$.

For the issue of computing randomized allocations there are two different notions of polynomial time computation. In a *random polynomial time implementation*, there is a randomized polynomial time algorithm that samples an allocation from the distribution D. In a *polynomial time implementation*, there is a deterministic polynomial time algorithm that lists all allocations in the support of D (implying in particular that the support contains at most polynomially many allocations), together with their associated probabilities.

2 Faithful Implementation

For additive valuations, there is a very useful lemma that greatly simplifies the design of BoBW allocations. We refer to it here as the *faithful implementation lemma*. The lemma (sometimes with slight variations) was previously stated and used in BoBW results [4, 5, 6, 1], and was used even earlier in approximation algorithms for maximizing welfare [8]. Restricted variants of it were introduced for scheduling problems [7], and were later used for allocation problems [3].

Lemma 1 *Let A^* be a fractional allocation of m items to n agents with additive valuations. Then there is a deterministic polynomial time implementation of A^*, supported only on allocations in which every agent gets value (ex-post) equal her ex-ante value (in the fractional allocation A^*), up to the value of one item. (For agent i, the corresponding one item is the item most valuable to i, among those items that are assigned to i under A^* in a strictly fractional fashion. Moreover, the values that the agent gets in any two allocations differ by at most the value of this single item.)*

Remark 2 *In [2], the statement of this lemma was augmented to also bound the size of the support of the distribution of the implementation. This aspect is omitted here.*

2.1 Historical context

In this section we provide some historical context as to the development of various components of Lemma 1.

Consider a fractional allocation A^* of m items to n agents with additive valuations. Denote the fractional allocation to agent i by A^*_i, with A^*_{ij} denoting the fraction of item j given to agent i in A^*. Let $M^f_i = \{j \mid 0 < A^*_{ij} < 1\}$

denote the set of items for which some positive proper fraction (neither 0 nor 1) is allocated to i, and let $f = \sum_{i \in \mathcal{N}} |M_i^f|$ denote the number of variables that are strictly fractional.

We consider generating a distribution over integral allocations from the fractional allocation A^* (a "rounding procedure"). We distinguish between three kinds of rounding:

- *Deterministic rounding.* Produces a single integral allocation.
- *Randomized rounding.* Produces a distribution over integral allocations.
- *Implementation.* Randomized rounding, where the expectation of the associated distribution is exactly A^*.

We consider two notions of polynomial-time algorithms for performing randomized rounding.

- *Randomized polynomial time.* There is a randomized polynomial time algorithm that samples an integer allocation from the associated distribution.
- *Deterministic polynomial time.* There is a deterministic polynomial time algorithm that lists all integral allocations in the support of the distribution, together with the associated probability of each allocation. In particular, this implies that the size of the support is upper bounded by some polynomial in n and m.

We list several *faithfulness* properties that may be associated with the rounding.

1. *Ex-post faithfulness*, which satisfy both of the following properties:

    (a) *Faithfulness from above.* In the rounded integral allocation A, every agent i gets a bundle of value at most her fractional value, up to the value of one of her fractionally allocated items. That is, $v_i(A_i) \leq v_i(A_i^*) + \max_{j \in \mathcal{M}_i^f}[v_i(j)]$.

    (b) *Faithfulness from below.* In the rounded integral allocation A, every agent i gets a bundle of value at least her fractional value, up to the value of one of her fractionally allocated items. That is, $v_i(A_i) \geq v_i(A_i^*) - \max_{j \in \mathcal{M}_i^f}[v_i(j)]$.

    For an implementation of a fractional allocation, Ex-post faithfulness follows from the following single property:

    - *Small spread.* For every agent i, the difference in values that i receives in any two rounded integral allocations is at most $\max_{j \in \mathcal{M}_i^f} v_i(j)$.

    We refer to a distribution over allocations as a *faithful implementation* of A^* if it is an implementation that satisfies small spread.

2. *Ex-ante faithfulness.* In the randomized rounding, every agent i gets in expectation value at least equal to her fractional value. $E[v_i(A_i)] \geq v_i(A_i^*)$. Observe that by definition, an implementation of the fractional allocation is ex-ante faithful.

Faithful rounding of fractional solutions has a long history, where in different times researchers added additional ingredients (from those mentioned above) that they wished to satisfy. We briefly mention a few past relevant works.

Independent randomized rounding has numerous applications for approximation algorithms. The rounding allocates each item to at most one agent, independently of the allocation of other items. That is, each item j is independently (from other items) allocated to at most a single agent, with each agent i getting item j with probability equal to A_{ij}^*. This procedure provides a randomized polynomial time implementation for the fractional allocation (and hence is ex-ante faithful), but it does not provide ex-post faithfulness guarantees.

Deterministic (polynomial time) rounding that is faithful from above was developed in [7] in the context of scheduling problems. For allocation problems, faithfulness from below is a more natural requirement, and this version was presented in [3]. A randomized polynomial time faithful implementation (showing that the small spread property holds and making explicit use it) was presented in [8]. A randomized polynomial time faithful implementation for a more general setting (referred to as a bi-hierarchy) was presented in [4]. Later work was concerned with deterministic (rather than randomized) polynomial time faithful implementations, with one approach described in [5], and a somewhat simpler approach presented in [1]. Lemma 1 summarizes the above discussion.

2.2 The Birkhoff – von Neumann theorem

Before proving Lemma 1, let us recall the Birkhoff – von Neumann theorem. We start with some background.

Let $G(U, V; E)$ be a bipartite graph where U is the set of left-hand side vertices, V is the set of right-hand side vertices, and E is the set of edges. A matching M in G is a subset of edges such that each vertex is incident with at most one edge of M. The matching is perfect if each vertex is incident with exactly one edge of M. A perfect matching may exist only if $|U| = |V|$, and we shall assume that $|U| = |V| = n$. A perfect matching exists if and only if Hall's condition holds: for every set $S \subseteq U$, the number of neighbors that S has (in V) is at least $|S|$. A perfect matching, if it exists, can be found in polynomial time. (More generally, a maximum size matching can be found in polynomial time, and likewise a maximum weight matching, if edges have weights.)

A fractional matching is an assignment of non-negative weights to the edges, where the sum of weights incident with each vertex is at most 1, and exactly 1 for a perfect fractional matching. The bipartite adjacency matrix A of a perfect fractional matching (the vertices of U index the rows, the vertices of V index the columns, and entry A_{ij} equals the weight of edge (i, j)) is doubly stochastic (it is a nonnegative matrix all whose row sums and all whose column sums equal 1).

We now state the Birkhoff – von Neumann theorem.

Theorem 3 *Every doubly stochastic matrix is a convex combination of permutation matrices. Equivalently, every perfect fractional matching in a bipartite graph can be decomposed into a weighted sum of perfect (integral) matchings. Moreover, such a decomposition can be found in polynomial time.*

For completeness, let us briefly recall the proof (or a proof) of Theorem 3. The setting is that of a bipartite graph G with n vertices on each side. The perfect fractional matching associates non-negative weights with the edges, where the sum of weights incident with each vertex is exactly 1. The goal is to find a collection $\mathcal{M} = M_1, M_2, \ldots$ of integral matchings, together with nonnegative coefficients $\lambda_1, \lambda_2, \ldots$, such that the fractional matching is the same as $\sum_i \lambda_i M_i$. That is, the weight of every edge e in G is equal to the sum of coefficients of those matchings that contain e.
Proof. Consider the graph G' that contains only those edges of G of positive weight. Observe that G' satisfies Hall's condition. Namely, for every set S of vertices on the left hand side, the number of neighbors that S has on the right side is at least $|S|$. (Otherwise, in G, at least one neighbor of S would need to be incident with weight larger than 1.) Hence G' contains a perfect matching. Find in G' a perfect matching M_1 (this can be done in polynomial time), and let ϵ be the smallest weight of an edge of M in G. Put M_1 in $\mathcal{M}$ with weight $\lambda_1 = \epsilon$, and reduce in G the weight of every edge of M_1 by ϵ. The number of edges of positive weight in G decreases by at least 1, and in the new G, every vertex is incident with weight exactly $1 - \epsilon$. The above argument can be repeated, extracting the matchings of $\mathcal{M}$ and their associated coefficients one by one. After at most n^2 repetitions, G has no more positive edges, and the procedure ends. ∎

Observe that we may view a perfect fractional matching in a bipartite graph as a fractional allocation, where the left hand side vertices are the agents and the right hand side vertices are the items. Hence Theorem 3 provides a polynomial implementation of this fractional allocation by integral allocations. In this respect, Theorem 3 is a special case of Lemma 1.

2.3 Proof of Lemma 1

We now prove Lemma 1 (based on a proof given in [1]).
Proof. We reduce the setting of Lemma 1 to that of the Birkhoff – von Neumann theorem, showing how we can take the fractional allocation A^* and generate from it a distribution over matchings of "clones" of each agent, that can be use to generate a distribution over allocations that is a faithful implementation of A^*. For every agent i we do the following. Let $f_i = \sum_j A^*_{ij}$ denote the total sum of fractions of items (not their values) received by i under A^*. We replace i by $\lceil f_i \rceil$ *clones* $c_i^1, \ldots, c_i^{\lceil f_i \rceil}$ as follows. Sort all items in order of decreasing v_i value. This gives a priority order for the following sequential "eating" process.

The clones of i "eat" the fractional allocation of i, where each clone in its turn consumes one unit of the fractional allocation (starting consuming only after the prior clone completed consuming), where the unit is chosen according to the priority order. The last clone might have less than a single unit to consume.

Having done the above for all agents, we now have a fractional matching between clones and items. This is not a perfect fractional matching (the last clone of an agent may consume less than one item), but the Birkhoff – von Neumann theorem still applies (e.g., one can add dummy clones and items as needed so as to complete the instance to a perfect fractional matching on a larger bipartite graph). Hence we can decompose the fractional matching into integral matchings. In every integral matching, every agent gets the items received by her clones.

Ex-post faithfulness follows from the fact that for every agent i, in every integral allocation, each of i's clones (except for perhaps the last one) receives one item. Let $S_{i,\max}$ ($S_{i,\min}$, respectively) be the set of items obtained by taking for each of i's clones the highest priority (lowest priority, respectively) item that the clone may possibly receive. Then every allocation that agent i may receive has value in the range $[v_i(S_{i,\min}), v_i(S_{i,\max})]$. Observe that $v_i(S_{i,\min}) \geq v_i(S_{i,\max}) - \max_{j \in \mathcal{M}_i^f} v_i(j)$. This last statement can be verified by removing the most valuable item (that of clone 1) from $S_{i,\max}$, and then using the fact that for every $j \leq 1$, the item of clone j in $S_{i,\min}$ is at least as valuable as the item of clone $j+1$ in $S_{i,\max}$. This established the small spread property, which implies ex-post faithfulness.

The above provides a deterministic polynomial time implementation of A^* as a distribution D over polynomially many allocations $A^1, A^2, \ldots A^\ell$, where every allocation in the support is ex-post faithful. ■

2.4 Applications to BoBW results and uses of linear programming

Using Lemma 1, one trivially gets the following BoBW result,

Proposition 4 *For allocation of indivisible goods to agents with additive valuations, there is a deterministic polynomial time implementation of a fractional allocation that is ex-ante proportional, and the implementation is supported on allocations that are (ex-post) Prop1.*

Proof. Consider the *uniform fractional allocation*, that assigns a fraction of $\frac{1}{n}$ of every item to every agent. It is ex-ante proportional (and also *envy-free*, a fairness notion not discussed here), as all agents get the same fractional allocation. Applying Lemma 1, it is implemented in deterministic polynomial time by allocations that are Prop1. ■

Lemma 1 can be combined with linear programming to give even stronger results. The idea is to use linear programming to select a fractional allocation which is more favorable than the uniform fractional allocation, and provide a faithful implementation of that allocation. We provide a few examples of how this is done.

In all cases, the input to the linear program (LP) is the additive valuation functions of the agents. The variables of the LP are x_{ij} specifying the fraction of item j allocated to agent i. An additional variable z denotes the objective function that we seek to optimize.

We assume that each valuation function v_i is scaled so that $v_i(\mathcal{M}) = n$, and so the proportional share of each agent is 1. Recall that we also assume that all items are *goods*, namely, $v_i(j) \geq 0$ for every agent i and item j. (Items that are not goods are referred to as *chores*.) In this case, we may desire to give every agent more than her proportional share ex-ante, if possible. This desire can be interpreted in several different ways. One is in a min-max sense, making the least happy agent as happy as possible. This gives rise to the following LP, referred to as LP1.

maximize z subject to:

1. $x_{ij} \geq 0$ for every agent i and item j. (Agents cannot receive negative fractions of items.)

2. $\sum_i x_{ij} \leq 1$. (The total fraction of item j that is allocated does not exceed 1.)

3. $\sum_j x_{ij} \cdot v_i(j) \geq z$ for every agent i. (Every agent receives a fractional bundle of total value at least z.)

Observe that all constraints above are linear in their variables, and so is the objective function. Hence, this is a linear program. As linear programs can be solved in polynomial time, a fractional allocation maximizing z can be found in polynomial time. The value of z is at least 1 (as the uniform fractional allocation is a feasible solution to the LP), but can potentially be much higher (up to n, if agents desire disjoint sets of items), and hence leads to a BoBW result that is stronger than that of Proposition 4.

Another interpretation of the desire to improve over the proportional share is in an aggregate sense, maximizing *welfare* (the sum of values received by all agents), conditioned on each agent receiving at least her proportional share. This gives rise to the following LP, referred to as LP2.

maximize z subject to:

1. $x_{ij} \geq 0$ for every agent i and item j. (Agents cannot receive negative fractions of items.)

2. $\sum_i x_{ij} \leq 1$. (The total fraction of item j that are allocated do not exceed 1.)

3. $\sum_j x_{ij} \cdot v_i(j) \geq 1$ for every agent i. (Every agent receives a fractional bundle of total value at least her proportional share.)

4. $\sum_i \sum_j x_{ij} \geq z$. (The total welfare is at least z.)

Observe that the fractional allocation found by LP2 is necessarily fractionally Pareto Optimal. As such, for every implementation of it, every integral allocation in the support of the implementation is also fractional Pareto Optimal. This gives the following strengthening of Proposition 4.

Proposition 5 *For allocation of indivisible goods to agents with additive valuations, there is a deterministic polynomial time implementation of a fractional allocation that is ex-ante proportional and fPO, and the implementation is supported on allocations that are (ex-post) Prop1 and fPO.*

Remark 6 *An alternative proof (taken from [5]) for Proposition 5 achieves additional desirable properties (such as ex-ante envy-freeness) beyond those specified in Proposition 5. The* Nash Social Welfare (NSW) *of allocation* $A = (A_1, \ldots, A_n)$ *is* $\left(\prod_{i \in \mathcal{N}} v_i(A_i)\right)^{\frac{1}{n}}$. *In case of fractional allocations, we use the notation fNSW. Selecting a fractional allocation that maximizes fNSW is not a solution to a linear program (the fNSW objective is not linear), but such an allocation can be found in polynomial time nevertheless (details omitted). Using Lemma 1 one can implement a fractional allocation that maximizes fNSW. Such a fractional allocation has (ex-ante) desirable properties that go beyond giving every agent at least her proportional share and being fPO.*

3 Homework

The combination of linear programming and (parts of) Lemma 1 has previously been used in order to obtain various approximation algorithms for NP-hard optimization problems. The homework presents two such cases. In both cases, you may use without proof Lemma 1 and the fact that LPs can be solved in polynomial time.

3.1 Scheduling so as to minimize makespan

The input is a set of n machines, a set of m jobs, a threshold t, and nonnegative processing times p_{ij} for every machine i and job j, specifying how long it would take to process job j on machine i. The goal is to allocate each job to a single machine (though the same machine can receive many jobs) so that no machine receives jobs with total processing time (on that machine) larger than t. We refer to such an allocation as a schedule of *makespan* at most t.

Give a polynomial time algorithm that given such an input instance either proves that no schedule with makespan at most t exists, or finds a schedule of makespan at most $2t$.

Remark 7 *The factor 2 approximation for makespan was first proved in [7]. It is still not known whether there is a polynomial time algorithm with an approximation ratio better than 2 for this problem.*

3.2 Maximizing welfare for budget additive agents

For the problem of allocating indivisible items, the valuation function v_i of an agent i is said to be *budget additive* if $v_i(e) \geq 0$ for every item e, and if there is some *budget* B_i such that for every set S of items, $v_i(S) = \min[B_i, \sum_{e \in S} v_i(e)]$. Given the budget additive valuation functions $v_1, \ldots, v_n$ of n agents, the goal is to find an allocation $A = (A_1, \ldots, A_n)$ that maximizes welfare, namely, maximizes $\sum_i v_i(A_i)$. The maximum welfare problem is easily solvable for additive valuations (give each item to the agent that values it most), but is NP-hard for budget additive valuations.

Design a polynomial time algorithm for allocating indivisible goods to agents with budget-additive valuations, that outputs an allocation whose welfare is at least a $\frac{3}{4}$ fraction of the maximum welfare.

An additional lemma that you may use is the following.

Lemma 8 *Let X be a nonnegative random variable with expectation μ, and supported only on values in some range $[a, b]$ (with $0 \leq a \leq \mu \leq b$). Let B satisfy $B \geq \max[\mu, b - a]$. Consider the random variable Y whose value is $Y = \min[X, B]$. (That is, one draws a random value for X, and if this value is larger than B then it is replaced by B.) Then the expectation of Y satisfies $E[Y] \geq \frac{3}{4}\mu$.*

(A hint for those attempting to prove Lemma 8: fixing μ, a, b, B, first show that the worst case is when the distribution of X is supported only on a and b.)

Remark 9 *The factor $\frac{3}{4}$ approximation was proved in [8]. It is not known whether there is a polynomial time algorithm with an approximation ratio better than $\frac{3}{4}$ for this problem.*

References

[1] Haris Aziz. Simultaneously achieving ex-ante and ex-post fairness. In *International Conference on Web and Internet Economics*, pages 341–355. Springer, 2020.

[2] Moshe Babaioff, Tomer Ezra, and Uriel Feige. Best-of-both-worlds fair-share allocations. *CoRR*, abs/2102.04909, 2021.

[3] Ivona Bezáková and Varsha Dani. Allocating indivisible goods. *ACM SIGecom Exchanges*, 5(3):11–18, 2005.

[4] Eric Budish, Yeon-Koo Che, Fuhito Kojima, and Paul Milgrom. Designing random allocation mechanisms: Theory and applications. *American economic review*, 103(2):585–623, 2013.

[5] Rupert Freeman, Nisarg Shah, and Rohit Vaish. Best of both worlds: Ex-ante and ex-post fairness in resource allocation. In *EC '20: The 21st ACM Conference on Economics and Computation, Virtual Event, Hungary, July 13-17, 2020*, 2020.

[6] Daniel Halpern, Ariel D. Procaccia, Alexandros Psomas, and Nisarg Shah. Fair division with binary valuations: One rule to rule them all. In *Web and Internet Economics - 16th International Conference*, 2020.

[7] Jan Karel Lenstra, David B Shmoys, and Éva Tardos. Approximation algorithms for scheduling unrelated parallel machines. *Mathematical programming*, 46(1):259–271, 1990.

[8] Aravind Srinivasan. Budgeted allocations in the full-information setting. In *Approximation, Randomization and Combinatorial Optimization. Algorithms and Techniques*, pages 247–253. Springer, 2008.