From Cost Sharing Mechanisms to Online Selection Problems

Adam N. Elmachtoub

Operations Research Center, Massachusetts Institute of Technology, ane@mit.edu, http://ane.scripts.mit.edu/home/

Retsef Levi

Sloan School of Management, Massachusetts Institute of Technology, retsef@mit.edu, http://retsef.scripts.mit.edu/

We consider a general class of online optimization problems, called *online selection problems*, where customers arrive sequentially, and one has to decide upon arrival whether to accept or reject each customer. If a customer is rejected, then a rejection cost is incurred. The accepted customers are served with minimum possible cost, either online or after all customers have arrived. The goal is to minimize the total production costs for the accepted customers plus the rejection costs for the rejected customers. These selection problems are related to online variants of offline prize collecting combinatorial optimization problems that have been widely studied in the computer science literature. In this paper, we provide a general framework to develop online algorithms for this class of selection problems. In essence the algorithmic framework leverages any *cost sharing mechanism* with certain properties into a poly-logarithmic competitive online algorithm for the respective problem; the competitive ratios are shown to be near-optimal. We believe that the general and transparent connection we establish between cost sharing mechanisms and online algorithms could lead to additional online algorithms for problems beyond the ones studied in this paper.

1. Introduction   The literature in supply chain, logistics, and network design theory provides many streamlined optimization models where the goal is to satisfy a deterministic or stochastic sequence of demand at minimum cost. More recent practice and research trends in have led to broader models that consider decisions on the supply side as well as on the demand side. More specifically, the customers to which one responds and commit to are not entirely exogenous parameters, but may be influenced by endogenous decisions such as pricing, promotions, and other strategic

marketing-based factors. In particular, a supplier or provider should strive to optimally match demand to the supply chain's production capabilities or the network's infrastructure. A fundamental aspect of this issue is the choice of customers to which the supplier commits to serving, and how these customers are implicitly, or explicitly, chosen. For example, the decision whether to serve a customer heavily depends on the customer's willingness to pay, as well as the costs expected to be incurred if the customer is satisfied. These decisions are often very challenging since at the time of the decision it is not clear what customers will arrive in the future. Moreover, in many setting there are economies of scale that create dependencies between the cost incurred by satisfying different customers.

In this work, we study a general class of *online customer selection* models that capture these challenges and tradeoffs in many important supply chain management, network design, and logistics areas. These models also capture online variants of many well-known and studied offline *prize-collecting problems*. For example, consider the facility location problem, where there a set of customers that require service and one must decide on the optimal set of facilities to open. The total cost is the cost to open the facilities plus the service cost for each customer to receive service from the nearest facility. In the facility location problem with online customer selection, customers arrive one-by-one and reveal their location. One must decide in real-time whether to accept and commit to serving the customer, or reject and incur a customer-specific rejection cost. Solving the facility location problem for the accepted customers is done either online or after all customers have arrived.

In this paper, we develop a general framework to design online algorithms for problems with online customer selection that can be applied under very little or no knowledge on future demands. The performance of the algorithms is analyzed by comparing them to the optimal solution of the respective *offline* variant, where the assumption is that the decision maker has fully upfront information on the future customers and their demands. This is known as the *competitive ratio* of the algorithm, i.e., the worst case ratio between the costs of the online algorithm and the optimal offline algorithm (Albers [1]).

The online algorithmic framework that we develop in this work is based on a novel use of *cost sharing mechanisms* for the respective offline problems (without customer selection). Cost sharing is an important and well-studied concept in economics and computer science, whereas one has to decide how to share the service (or production) costs amongst the players that received service (Moulin and Shenker [35], Jain and Mahdian [26]). These problems arise when there are multiple players that want to pay for service or goods, but the production costs have economies of scale. In other words, the cost of serving every player separately is far greater than the cost of serving them all simultaneously due to the economies of scale. For a given production (or service) cost function, a cost sharing mechanism determines how much each player should pay to be served and distributes the economies of scale in a fair manner. In this paper, we obtain general results that show how to leverage any cost sharing mechanism for an offline problem with certain (rather general and commonly used) properties into an online algorithm for the corresponding online selection problem. For example, for the facility location problem we use the cost sharing mechanism of Pál and Tardos [37] to obtain an online algorithm for the facility location problem with online customer selection. In fact, the competitive ratio of the online algorithm is directly characterized by certain characteristics of the respective cost sharing mechanism. This leads to a general framework that not only leverages existing cost sharing mechanisms, but also will transparently leverage any future result in this area. Moreover, for many of the problems that we study, it is shown that these competitive ratios are close to best possible by *any* online algorithm.

1.1. Models. In this work we study a large class of online selection variants to core supply chain management, logistics and network design problems, such as the *economic lot sizing* problem, the *facility location* problem and the *minimum Steiner tree* problem. In these online customer selection variants, decisions are made in two phases. First, there is a *selection phase*, in which the supplier has to decide in *real-time* (i.e., upon customer arrivals) whether to accept or reject each arriving customer. Customers have different willingness to pay that is captured through an associated revenue, and may have different requirements with respect to quantities desired, due

dates, and locations. After (or while) the selection decisions are made, there is a *production phase*, in which the accepted customers are served with minimum possible production cost. The goal is to minimize the total cost of lost revenue from rejected customers plus the cost of satisfying the accepted customers. Most of the previous related work assumed that all customers are known in advance when the accept/reject decisions are made; this is called *offline customer selection* (Goemans and Williamson [21], Charikar et al. [14], Geunes et al. [18]). In contrast, we consider a much more realistic setting, where selection decisions are made adaptively over time with *no* information on the future customers and their demands. This lack of information about the future is a more realistic assumption in many practical settings. However, it adds significantly to the complexity of the decision-making problem, particularly in settings where the production costs have economies of scale. The types of production costs we consider are the cost of economic lot sizing, facility location, and network design problems.

The online customer selection models studied in this work attempt to capture real-life operational situations of a make-to-order supplier or service provider. Typically, decisions are broken up into phases (weeks, quarters, years). In each phase, the supplier receives customer orders (requests) due in some time period in the next phase, and needs to instantaneously decide which customers to serve as their requests arrive. During the same phase, the supplier is also serving the customers that were accepted in the previous phase with minimum production cost. In many of these settings, it is often extremely challenging to form a reliable distribution of future customers and their demands. The challenge stems from market volatility, lack of reliable data, and the fact that customers have very different needs for customized orders (or service) and associated revenues. In light of these issues, the assumption that the supplier has minimal knowledge about future customer arrivals could lead to more robust policies. Our model and algorithms can also help the supplier suggest an appropriate price to the customer that would ensure profitability for the supplier.

1.2. Contributions. In this paper, we develop a general class of online algorithms called *FairShare* to make online customer selection decisions. We show that FairShare performs well for

a variety of inventory management, facility location, and network design problems with online customer selection. As previously mentioned, the performance is measured by the notion of competitive ratio, which is widely used in many online optimization settings. An online algorithm with a competitive ratio of c means that, even in the worst-case scenario, the cost of the online algorithm is at most c time the cost of the optimal offline solution.

The FairShare algorithm is based on repeatedly simulating a *cost sharing mechanism* for the respective production problem. In this type of mechanism, N "players" each submit a "bid" representing how much they would like to be served. Then, the mechanism decides which set of players to serve and how much each player should pay according to a *cost sharing method*. This cost sharing method is a proxy for how much each customer contributes to the overall production cost. When FairShare simulates a cost sharing mechanism, all previously observed customers are the players and their associated revenues are the bids. If the simulated cost sharing mechanism decides to serve the customer that just arrived, then FairShare accepts that customer, and otherwise rejects. We show that the competitive ratio of FairShare is directly linked to certain characteristics of the cost sharing method. Therefore, this allows us to leverage many existing results for cost sharing mechanisms in order to generate online selection algorithms with strong performance guarantees.

Specifically, desirable cost sharing methods satisfy several nice properties such as *competitiveness*, *cost recovery*, and *cross-monotonicity*. Competitiveness implies that the sum of the cost shares assigned to the selected players is no more than the total cost of service for those players, while cost recovery implies that the sum of the cost shares covers a large fraction of the actual cost to serve the accepted players. Cross-monotonicity means that the cost shares of each player decreases as more players receive service. These three properties together imply that cost sharing methods provide a very good way to *share* the costs of production *fairly* across customers so that good selection decisions can be made. We also rely on the *summability* of the cost sharing method, described by Roughgarden and Sundararajan [39], which turns out to characterize how much FairShare will regret rejecting customers. We believe that our use of game theory mechanisms in an online optimization setting has potential for many more applications. Finally, we obtain lower bounds

on the best possible competitive ratio that depend on the logarithm of the number of customers. This shows that FairShare achieves competitive ratios (i.e., upper bounds) that are very close to optimal, which provides even stronger evidence of the intrinsic connection between cost sharing mechanisms and online selection problems.

1.3. Literature Review. Earlier work on the prize-collecting traveling salesman problem and the prize-collecting Steiner tree problem studied by Bienstock et al. [8] and Goemans and Williamson [21], respectively, falls into our offline customer selection framework. In Charikar et al. [14], an approximation algorithm is given for the facility location problem with offline customer selection. Some models considered offline *market* selection, where a market is a sequence of demands requested by a customer over time (or in multiple locations) that must be either fully accepted or rejected. van den Heuvel et al. [44] shows that it is NP-hard to approximate the profit maximization variants of these models within any constant. This motivates the focus on cost minimization variants that we and others have studied. In Geunes et al. [18], a general linear programming rounding framework that provides approximation algorithms for several production planning and logistics problems with offline market selection. In contrast, the problems in this work consider markets that consist only of a single period (or location), but the market/customer selection decisions are made online without any information about the future. A more general area of literature can be found in Slotnick [42].

There has also been a recent stream of literature on the use online optimization for operational planning. In van den Heuvel and Wagelmans [45], the competitive ratio of the online economic lot sizing problem *without* customer selection is shown to have a lower bound of two, matching the best known guarantee achieved in Axsater [3]. More general single item models are considered in Wagner [47]. In Buchbinder et al. [11], an online primal-dual algorithm is proposed for a make-to-order variant of the online joint replenishment problem with a competitive ratio of three. The work of Fotakis [17] provides a lower and matching upper bound depending on the number of customers for the online facility location problem. Meyerson [33] provides randomized algorithms for a variant of

the online facility location problem that gives a constant competitive ratio. In Ball and Queyranne [5], booking policies are found that achieve small competitive ratio for online revenue management problems. Finally, Jaillet and Lu [24] propose algorithms for the online traveling salesman problem with online customer selection. The ideas used in all these papers are very different than the ones used in this work.

The area of cooperative game theory, which focuses on how to fairly allocate costs among players, has gained recent interest in operations management (see Cachon [13], Nagarajan and Sosic [36], Bhaskaran and Krishnan [7], Kim and Netessine [28]). Of main interest to this paper are the cost sharing mechanisms for inventory management (Xu and Yang [48]), facility location (Pál and Tardos [37]) and Steiner tree (Jain and Vazirani [27]) problems. Roughgarden and Sundararajan [39] developed a notion called *summability*, which was used to study the social welfare of the previously mentioned mechanisms. Our paper is the first to leverage cost sharing mechanisms directly in an online optimization setting.

In previous recent work, Elmachtoub and Levi [16] considered a special class of online customer selection problems, where the rejection cost per unit was the same for all customers. Under this assumption, they were able to obtain online algorithms with constant competitive ratios. The algorithms they used were based on variants of re-optimization that ignored previously made decisions. However, the uniform rejection cost assumption does not capture situations where the willingness to pay among customers may vary (i.e., different associated revenues among customers) or where prices may be changing over time. Moreover, we show that if the rejection costs vary with each customer, no constant competitive ratio guarantees exist for the applications we consider.

The remainder of this paper proceeds as follows. In Section 2, we describe a general model for online customer selection problems along with some useful notation. In Section 3, we describe the FairShare Algorithm and provide a general analysis of its performance. In Section 4, we give several applications with polylogarithmic competitive ratios. In Section 5, we provide a lower bounds on the competitive ratio for the facility location and economic lot sizing problems with online customer

8

Elmachtoub and Levi: *From Cost Sharing in Online Selection Problems*
Article submitted to *Mathematics of Operations Research*; manuscript no. (Please, provide the mansucript number!)

selection (minimum Steiner tree lower bound is in Appendix). In Section 6, we discuss an extension to online market selection. In Section 7, we offer some concluding remarks and future research directions.

2. General Model and Notation The models studied in this paper involve decisions that are made in two phases. First, there is a *selection phase* in which customers arrive sequentially in an online manner. In *stage k* of the selection phase, customer k arrives with requirements I_k and rejection cost r_k. Requirements (information) I_k may include demand quantities, due dates, and locations needed by the customer. After customer k arrives, the supplier needs to decide whether to accept or reject customer k using only the information regarding the first k customers. If customer k is rejected, a cost of r_k is incurred, which represents the lost revenue from that customer. The selection phase completes when the supplier stops observing new customers. At this point the customers that arrived have been partitioned into accepted and rejected sets denoted by $\mathcal{A}$ and $\mathcal{R}$, respectively.

The second phase is the *production phase*, where the accepted customers are served accordingly to meet their requirements. For a set of customers T, $P(T)$ denotes the minimum possible production cost to serve the customers in T. The function $P(T)$ typically represents the cost of an optimal solution to a minimization problem. The production cost for the accepted set of customers $\mathcal{A}$ is then denoted by $P(\mathcal{A})$. We make the natural assumptions that $P(T)$ is nondecreasing in T and $P(\emptyset) = 0$. The overall goal is to minimize the total production costs of the accepted customers plus the rejection costs of the rejected customers. This two phase problem is generally referred to as an *online customer selection* problem. In Section 4 we will describe specific applications of the model by specifying definitions for I_k, r_k, and $P(\cdot)$. In addition, we also consider the possibility where the two phases overlap and one needs to solve the production problem online as well.

We now outline convenient notation used throughout the paper. Let N be the number of customers that arrived (unknown a priori), U be the full set of customers $\{1, \ldots, N\}$ and U_k be the first k customers $\{1, \ldots, k\}$, implying that $U = U_N$. When referring to the final stage N, the subscript

may be dropped for simplicity. The rejection cost of a subset of customers $T \subseteq U$ is defined as $R(T)$, i.e., $R(T) = \sum_{k \in T} r_k$. The notation $\mathcal{A}_k$ and $\mathcal{R}_k$ denote the customers that were accepted and rejected, respectively, by the online algorithm through the first k stages. Note that $\mathcal{A}_k \cup \mathcal{R}_k = U_k$, $\mathcal{A}_k \cap \mathcal{R}_k = \emptyset$, $\mathcal{A}_{k-1} \subseteq \mathcal{A}_k$, and $\mathcal{R}_{k-1} \subseteq \mathcal{R}_k$ for all k.

If all information $I_1, \ldots, I_N$ and $r_1, \ldots, r_N$ is known upfront, then the *offline customer selection problem* is defined as $\mathrm{OPT}(U) = \min_{A \subseteq U} P(A) + R(U \backslash A)$. Let $\mathcal{A}_k^*$ and $\mathcal{R}_k^*$ denote an optimal pair of accepted and rejected sets in the offline problem $\mathrm{OPT}(U_k)$. Note that $\mathcal{A}_k^* \cup \mathcal{R}_k^* = U_k$ and $\mathcal{A}_k^* \cap \mathcal{R}_k^* = \emptyset$ for all k, but monotonicity does not necessarily hold since an offline solution may change its selection decisions as the stages progress.

The optimal offline cost through stage k is denoted by $C^*(U_k)$, which can be expressed as $C^*(U_k) = P(\mathcal{A}_k^*) + R(\mathcal{R}_k^*)$. The value $C(U_k)$ denotes the total cost incurred by the online algorithm through stage k, i.e., $C(U_k) = P(\mathcal{A}_k) + R(\mathcal{R}_k)$. Using these definitions, it follows that for any online algorithm and any stage k, $C^*(U_k) \leq C(U_k)$. We note that another natural objective to use is profit, i.e., $R(\mathcal{A}_k) - P(\mathcal{A}_k)$. Although this objective is equivalent in an offline setting , i.e., it results in the same optimal offline solution, approximating profit in an online setting is very challenging due to the mixed sign objective. In fact, even in an offline setting approximating profit within any factor is NP-hard for many customer selection problems (van den Heuvel et al. [44]).

The performance of an online algorithm is evaluated using the notion of *competitive ratio*. An algorithm has a competitive ratio of d and is called d-competitive if $C(U) \leq dC^*(U)$ for any online sequence of customers U and their respective characteristics. In other words, the cost of the algorithm is guaranteed to be at most d times the cost of an optimal offline solution for *any* customer sequence.

3. Algorithm Our algorithm is based on the repeated simulation of specific cost sharing mechanisms. Suppose we are given a set of players U and a cost function $P(\cdot)$. A *cost-sharing mechanism* is a mechanism that first collects bids from each player in U, and then decides which subset of players to serve. In addition, the mechanism assigns a cost share (price) to each serviced

player. If the mechanism chooses to serve $T \subseteq U$, then we let $\chi(i, T)$ denote the cost share assigned to player $i \in T$ for service. We refer to the function χ as a cost sharing scheme (or method), which can possibly take on some of the properties discussed later on. In the next section we describe several useful cost sharing methods for various production cost functions of interest.

Given a set of customers T, a cost-sharing scheme χ, and a scalar c, we let $M(\chi, T, c)$ denote the set of customers output by the following procedure, known as a *Moulin mechanism* (Moulin [34]). A Moulin mechanism is a class of cost sharing mechanisms where every player begins by receiving service, and players are iteratively removed if their cost share is higher than their (scaled) bid. The mechanism terminates when every remaining player has a cost share less than or equal to his (scaled) bid, which we denote by r_i for player i for convenience.

Moulin Mechanism: $M(\chi, T, c)$

1. Collect bids r_i from each player $i \in T$.
2. If $cr_i \geq \chi(i, T)$ for all $i \in T$, halt and output T.
3. Else, let $j \in T$ be a player with $cr_j < \chi(j, T)$. Set $T := T \backslash \{j\}$.
4. Go to Step 2.

Now we can define our general online customer selection algorithm, which we call FairShare, according to the following procedure. FairShare relies on a cost sharing method χ and a scalar c, where c is a cost scaling parameter chosen to optimize performance. The cost sharing method χ must be specific for the production cost function $P(\cdot)$ being considered in the application, and will act as a proxy for how much each customer is contributing to the overall production cost of the accepted customers. For example, we will use the facility location cost sharing method of Pál and Tardos [37] for running the FairShare on the facility location problem with online customer selection. For the current customer k, the FairShare algorithm is simulating a Moulin mechanism with χ as the cost sharing method, the entire set of observed customers so far U_k as the players, and their respective rejection costs as their bids. If the simulated mechanism accepts customer (player) k, then so does FairShare and vis versa.

> **FairShare** Accept current customer k if and only if $k \in M(\chi, U_k, c)$.

Clearly the FairShare algorithm converges in at most N iterations since there is at least one person removed in every iteration except for the very last one. Furthermore, since all of the cost sharing schemes we consider can be computed efficiently in polynomial time, then FairShare is generally an efficient algorithm. We now define some desirable properties for χ. A cost sharing scheme χ is

1. *cross-monotonic* if $\chi(i, S) \geq \chi(i, T)$ for all $i \in S \subseteq T \subseteq U$.

2. (β, γ)-*budget balanced* if $\frac{P(T)}{\gamma} \leq \sum_{i \in T} \chi(i, T) \leq \beta P(T)$

3. α-*summable* for a monotonically increasing function $\alpha(\cdot)$ if $\sum_{l=1}^{|T|} \chi(i_l, T_l) \leq \alpha(|T|) P(T)$ for every ordering σ of U and every set $T \subseteq U$, where T_l denotes the set of the first l demands in T and i_l denotes the lth demand of T (with respect to σ).

The cross-monotonicity property simply states that if the number of players receiving service grows, then the cost share of each individual will not increase. In other words, if we remove a player, then the cost share of all remaining players can only increase. In the mechanism design setting, if χ is cross-monotonic, then the Moulin mechanism will induce truthful bidding (Moulin and Shenker [35]).

The (β, γ)- budget balanced property ensures that the sum of the cost shares is no more than β times the associated cost, but also recovers at least $1/\gamma$ of the cost. A $(1, \gamma)$-budget balanced is said to be *competitive* while a $(\beta, 1)$-budget balanced mechanism is said to be *no-deficit*. Although χ is providing a specific approximation to $P(\cdot)$, we use the exact value of $P(\cdot)$ computing the values of $C^*(\cdot)$ and $C(\cdot)$.

Finally, the α-summable property is a measure of the efficiency (social welfare), of the Moulin mechanism, as described by Roughgarden and Sundararajan [39]. To see this, pick an arbitrary order σ of $T \subseteq U$, and let each player l's valuation, and therefore bid, be $\chi(l, T_l) - \epsilon$ for some arbitrarily small ϵ. If we run the Moulin mechanism $M(\chi, T, 1)$, then no player will be chosen, since

they will be eliminated in reverse order by construction. Therefore the total cost of the system will be roughly $\sum_{l=1}^{|T|} \chi(i_l, T_l)$, where the optimal cost would be just $C(U)$. If σ is chosen adversarily, then $\alpha(\cdot)$ represents the worst possible situation that we can construct using the previous example. (A more rigorous analysis is in Roughgarden and Sundararajan [39].) We note that contrary to β and γ, α is typically not a scalar but rather a function that depends on the number of customers being served.

Given a cost sharing scheme χ that satisfies these three properties, we can intuitively explain why we can expect the FairShare algorithm to perform well. First, the cross-monotonicity implies that once a player is accepted by FairShare, he will always be accepted in every simulated output of the Moulin mechanism after his arrival. Thus, the current set of accepted customers is always a subset of the current output of the Moulin mechanism, whose costs can be bounded by the other two properties. In essence, cross-monotonicity implies that χ is a *fair* way to *share* the costs among the customers, since as more customers are added, the less each customer is 'charged' for service. Due to the (β, γ)-budget balanced property, we can expect that the production costs for serving the accepted customers will be not too large. Finally, motivated by the previous example, the α-summability directly characterizes how much we will expect to incur in rejection costs. In the next section, we formally provide theoretical guarantees on the cost of the FairShare algorithm. We explicitly characterize the competitive ratio of FairShare as a function of α, β, γ, c, and N. In Section 4, we discuss specific cost sharing methods from the literature that yield relatively small competitive ratios.

3.1. Analysis   In this section, we will provide explicit bounds on both the production and rejection costs incurred by FairShare. We assume that we have a (β, γ)-budget balanced, α-summable, cross-monotonic, cost sharing scheme χ for the production cost function $P(\cdot)$. Remember that U_k refers to the first k customers that have arrived. We will let $\mathcal{A}_k^M = M(\chi, U_k, c)$ and $\mathcal{R}_k^M = U_k \backslash M(\chi, U_k, c)$ for a given choice of parameter c to be optimized later. A set with no subscript simply refers to the final customer N. We first prove a key lemma which says that the the

sets $\mathcal{A}_k^M$ are monotonically increasing in k. Intuitively, the reason for this is that as more customers are added, the better off everyone is due to the cross-monotonicity of χ. This will be key to prove our production bound in Lemma 2.

LEMMA 1.   *The accepted set of customers by $M(\chi, U_k, c)$ is monotonically increasing in k, i.e.,*
$$\mathcal{A}_1^M \subseteq A_2^M \subseteq \ldots \subseteq A_N^M.$$

Proof.   We will prove the lemma by induction. The base case is trivial since we start the problem with no customers. Assume $\mathcal{A}_1^M \subseteq A_2^M \subseteq \ldots \subseteq A_k^M$ and we want to show that $\mathcal{A}_k^M \subseteq \mathcal{A}_{k+1}^M$. Assume for contradiction that a customer $j \in \mathcal{A}_k^M$ was not in $\mathcal{A}_{k+1}^M$. In case there are multiple choices for j, we choose the j that was first removed by the Moulin mechanism for $M(\chi, U_{k+1}, c)$. Let S_j denote the set of players that had not been rejected by the Moulin mechanism for $M(\chi, U_{k+1}, c)$ just before j was removed. By construction, $j \in S_j$. Then we know that

$$cr_j < \chi(j, S_j) \leq \chi(j, \mathcal{A}_k^M) \leq cr_j.$$

The first inequality follows from the fact that j was removed by the mechanism $M(\chi, U_{k+1}, c)$ when the current remaining set at the time was S_j. The second inequality follows from the cross-monotonicity of χ and the fact that by definition $\mathcal{A}_k^M \subseteq S_j$ by construction. The last inequality follows from the fact that the set $\mathcal{A}_k^M$ was previously output by the Moulin mechanism. Thus, we arrived at a contradiction, which implies that each $j \in \mathcal{A}_k^M$ is also in $\mathcal{A}_{k+1}^M$, and thus $\mathcal{A}_k^M \subseteq \mathcal{A}_{k+1}^M$.

$\square$

Using this lemma, we can now prove a bound on the production costs of FairShare.

LEMMA 2.   *The production costs of the FairShare algorithm are $P(\mathcal{A}) \leq \gamma \left(cR(\mathcal{R}^*) + \beta P(\mathcal{A}^*) \right)$.*

Proof.   The production cost of serving the accepted customers is

$$P(\mathcal{A}) \leq P(\mathcal{A}^M)$$

$$\leq \gamma \sum_{k \in \mathcal{A}^M} \chi(k, \mathcal{A}^M)$$

$$\leq \gamma \left(\sum_{k \in \mathcal{A}^M \cap \mathcal{R}^*} \chi(k, \mathcal{A}^M) + \sum_{k \in \mathcal{A}^M \cap \mathcal{A}^*} \chi(k, \mathcal{A}^M \cap \mathcal{A}^*) \right)$$

$$\leq \gamma \left(\sum_{k \in \mathcal{A}^M \cap \mathcal{R}^*} \chi(k, \mathcal{A}^M) + \beta P(\mathcal{A}^M \cap \mathcal{A}^*) \right)$$

$$\leq \gamma \left(\sum_{k \in \mathcal{A}^M \cap \mathcal{R}^*} \chi(k, \mathcal{A}_k^M) + \beta P(\mathcal{A}^M \cap \mathcal{A}^*) \right)$$

$$\leq \gamma \left(\sum_{k \in \mathcal{A}^M \cap \mathcal{R}^*} cr_k + \beta P(\mathcal{A}^M \cap \mathcal{A}^*) \right)$$

$$= \gamma \left(cR(\mathcal{A}^M \cap \mathcal{R}^*) + \beta P(\mathcal{A}^M \cap \mathcal{A}^*) \right)$$

$$\leq \gamma \left(cR(\mathcal{R}^*) + \beta P(\mathcal{A}^*) \right).$$

The first inequality follows since Lemma 1 implies $\mathcal{A} \subseteq \mathcal{A}^M$ and $P(\cdot)$ is monotonic. The second inequality follows from χ being (β, γ)-budget balanced. The third inequality follows from cross-monotonicity of χ. The fourth inequality follows from χ being (β, γ)-budget balanced. The fifth inequality follows Lemma 1 which implies $\mathcal{A}_k^M \subseteq \mathcal{A}^M$ and the cross-monotonicity of χ. The sixth inequality follows from the fact that $\mathcal{A}_k^M$ is an output of the Moulin mechanism. The equality follows from the definition of $R(\cdot)$. The last inequality follows from the monotonicity of $R(\cdot)$ and $P(\cdot)$. $\square$

We now show that any subset of the rejected customers of FairShare will be rejected by the Moulin mechanism as well. Intuitively this is also due to the cross-monotonicity of χ since these customers were already being rejected when they were in consideration with an even larger set of customers. This result will be useful in proving our rejection cost bound in Lemma 4.

LEMMA 3. *For any subset of customers $Q \subseteq \mathcal{R}$, the Moulin mechanism for $M(\chi, Q, c)$ outputs the empty set.*

Proof. Assume for contradiction that the Moulin mechanism for $M(\chi, Q, c)$ outputs $T \neq \emptyset$. Let k be the last customer that arrived in T. Consider the first time a customer $j \in T$ was removed by $M(\chi, U_k, c)$ and let T_j denote the players remaining just before j was removed. Note since $k \in \mathcal{R}$ that j and T_j are well-defined. Then we know that

$$cr_j < \chi(j, T_j) \le \chi(j, T) \le cr_j.$$

The first inequality from the fact j was removed by the Moulin mechanism for $M(\chi, U_k, c)$ when the current remaining set at the time was T_j. The second inequality follows from the fact that $T \subseteq T_j$ by construction and the cross-monotonicity of χ. The last inequality follows from the fact that T is an output of the Moulin mechanism. Thus we arrived at a contradiction and conclude that T must be indeed empty. $\square$

Using the previous lemma, we can now obtain a bound the rejection costs of the FairShare algorithm.

LEMMA 4. *The rejection costs of the FairShare algorithm are* $R(\mathcal{R}) \le R(\mathcal{R}^*) + \frac{\alpha(N)}{c} P(\mathcal{A}^*)$.

Proof. Let $Q = \mathcal{R} \cap \mathcal{A}^*$ which is the set of customers that FairShare rejected but the optimal offline solution accepted. From Lemma 3, we know that $M(\chi, Q, c)$ returns the empty set. For each $j \in Q$, let Q_j be the remaining set of customers just before customer j was rejected in the Moulin mechanism for $M(\chi, Q, c)$. Then by construction we know that

$$cr_j < \chi(j, Q_j).$$

Then we can show that

$$
\begin{aligned}
R(\mathcal{R} \cap \mathcal{A}^*) = R(Q) & \\
& < \sum_{j \in Q} \frac{\chi(j, Q_j)}{c} \\
& \le \frac{\alpha(|Q|)}{c} P(Q) \\
& \le \frac{\alpha(N)}{c} P(\mathcal{A}^*).
\end{aligned}
$$

The first equality follows from the definition of Q. The first inequality follows from the previous inequality we derived. The second inequality follows from α-summability of χ. The last inequality follows from the monotonicity of $\alpha(\cdot)$ and $P(\cdot)$. Adding $R(\mathcal{R} \cap \mathcal{R}^*) \le R(\mathcal{R}^*)$ to both side of the inequality completes the proof. $\square$

Using the bounds we have obtained for the FairShare algorithm, we can now easily prove the general theorem below.

THEOREM 1. *Given a (β,γ)-budget balanced, α-summable, cross-monotonic cost sharing scheme χ for a production problem P with online customer selection, the FairShare algorithm is $\max(1+\gamma c, \frac{\alpha(N)}{c}+\gamma\beta)$-competitive.*

Proof.   We have that

$$C(U) = R(\mathcal{R}) + P(\mathcal{A})$$

$$\leq R(\mathcal{R}^*) + \frac{\alpha(N)}{c}P(\mathcal{A}^*) + P(\mathcal{A})$$

$$\leq R(\mathcal{R}^*) + \frac{\alpha(N)}{c}P(\mathcal{A}^*) + \gamma\left(cR(\mathcal{R}^*) + \beta P(\mathcal{A}^*)\right)$$

$$\leq \max(1+\gamma c, \frac{\alpha(N)}{c}+\gamma\beta)(R(\mathcal{R}^*)+P(\mathcal{A}^*))$$

$$= \max(1+\gamma c, \frac{\alpha(N)}{c}+\gamma\beta)C^*(U).$$

The first equality follows from the definition of $C(\cdot)$. The first inequality follows from Lemma 4. The second inequality follows from Lemma 2. The third inequality is combining terms. The last equality follows from the definition of $C^*(\cdot)$ which completes the proof.   □

The following corollary states what happens if we want to use an approximation algorithm or online algorithm for P.

COROLLARY 1. *If a d-approximation or d-competitive algorithm is used to compute $P(\cdot)$, then the FairShare algorithm is $\max(1+\gamma cd, \frac{\alpha(N)}{c}+\gamma\beta d)$- competitive.*

Finally, if the number of customers is known up to a constant factor, then c can be chosen to be $\Theta(\sqrt{\alpha(N)})$ which optimizes the bound in Theorem 1. In particular, knowing N approximately allows for a substantial improvement in the competitive ratio from $O(\alpha(N))$ to $O(\sqrt{\alpha(N)})$.

COROLLARY 2. *If N is known up to constant factors, and β and γ are scalars, then choosing $c = \Theta(\sqrt{\alpha(N)})$ makes the FairShare algorithm $O(\sqrt{\alpha(N)})$-competitive.*

4. Applications   We now describe several problems with online customer selection and then state the main results (see Table 4). The metric *Facility Location (FL)* problem is a well studied NP-hard problem. The goal is to serve a set of customers, each with a specified demand quantity and location, by opening a set of facilities. There are M potential facilities, indexed by $j = 1, \ldots, M$. The opening cost of facility j is f_j. Each unit of demand is served by the nearest open facility, and pays a service cost $c(j, k)$ if facility j serves customer k. We assume that c induces a metric (symmetric and satisfies triangle inequality) over the facilities and customers' demands. The goal is to serve all the customers so as to minimize the total facility costs plus service costs. The first constant factor approximation algorithm was given by Shmoys et al. [41], and the current best approximation factor is 1.49 due to Li [31]. When customers can be rejected, this is known as the *Facility Location Problem with Offline Customer Selection.* The first approximation algorithm was given by Charikar et al. [14], and the current best approximation factor of 1.52 is due to Li et al. [32]. In this paper, we focus on the *Facility Location Problem with Online Customer Selection.* When a customer k arrives online, we observe $I_k = (l_k)$, where l_k specifies the location of where customer k's demand is. The rejection cost is just r_k. The production cost function $P(T)$ is now the optimal cost of the FL problem for a given set of customers T. Pál and Tardos [37] provides a cost sharing scheme that is cross-monotonic and $(3, 1)$-budget balanced. Roughgarden and Sundararajan [40] show that this scheme is log-summable in the number of customers.

The *Economic Lot Sizing (ELS)* problem is a single item, single location, discrete time inventory model. There is a set of customers, each with a due date and single unit demand, that need to be served by a sequence of production orders over a planning horizon of T periods. Each order incurs a setup cost K. Each customer is served by exactly one order. If a customer with due date t is satisfied by an order at time $s < t$, then a per unit holding cost $h \geq 0$ is incurred for every period in inventory from period s to t. Similarly, if the demand is served by an order $s > t$ then a per unit per period backlogging cost $b \geq 0$. The objective is to minimize the total setup ordering cost plus holding and backlogging costs. The ELS problem can be solved efficiently via dynamic programming (Wagner and Whitin [46]). The *Economic Lot Sizing problem with Offline Customer*

18

Elmachtoub and Levi: *From Cost Sharing in Online Selection Problems*
Article submitted to *Mathematics of Operations Research*; manuscript no. (Please, provide the mansucript number!)

Selection can also be solved efficiently (Geunes et al. [19]). We will focus on the *Economic Lot Sizing problem with Online Customer Selection*. Customers arrive in an online manner and the supplier needs to make an immediate selection decision before new customers arrive. When a customer k arrives, he specifies $I_k = (t_k)$, where t_k is the due date for customer k. The rejection cost of customer k is r_k. The production cost $P(T)$ is then the optimal cost of the ELS problem on a subset of customers T. If the rejection cost per unit is fixed, then Elmachtoub and Levi [16] shows that there is a 3-competitive online algorithm. Xu and Yang [48] adapted the cost sharing scheme of Pál and Tardos [37] and found one that is cross-monotonic and $(\max(b/h, h/b), 1)$-budget balanced. Using the same proof as in Roughgarden and Sundararajan [40] for the Pál and Tardos [37] scheme, one can show that this scheme is log-summable in the number of customers.

The *Steiner tree* problem is a network design model where we are a given a graph $G = (V, E)$ and subset of nodes $S \subseteq V$ that needs to be connected with a tree. The cost of the tree is the sum of the costs of each edge used in the tree. The first approximation algorithm was given by Gilbert and Pollak [20], and the current best approximation factor of 1.39 is due to Byrka et al. [12]. Extensions of this problem include the Steiner forest problem (Jain [25]), the Single Source Rent or Buy (SSROB) problem (Talwar [43]), and the Multi-commodity Rent or Buy (MROB) problem (Kumar et al. [30]). (The Steiner forest problem requires one to connect a set of node pairs. The rent-or-buy problems assume that each edge is either paid per use or bought once for a fixed cost. The single commodity and multi-commodity variants correspond to the Steiner tree and Steiner forest problems, respectively.) When nodes can be rejected, this is typically called the *Prize-Collecting Steiner Tree* problem. The first approximation for this was due to Bienstock et al. [8], and the current best approximation factor of 1.99 is due to Archer et al. [2]. Directly related to these problems is the prize-collecting traveling salesman problem Balas [4]. We focus on the *Steiner Tree Problem with Online Customer Selection* where customers arrive online and the information for customer k is his node location. The production cost $P(T)$ is the optimal cost of the Steiner tree to serve the customers in T. Jain and Vazirani [27] provides a cost sharing scheme that is cross-monotonic and $(2, 1)$-budget balanced for the minimum spanning tree problem, which

can also be used for the Steiner tree problem. Roughgarden and Sundararajan [39] show that this scheme is $\log^2$-summable in the number of customers.

The table below summarizes the results based on the ideas in this paper. The first column denotes the production cost problem along with the citation of the cost sharing method that we use in FairShare. The second column displays the summability function α for the corresponding problem along with the paper that showed this. The third and fourth columns show the β and γ factors that were shown with the corresponding cost-sharing method paper. The fifth column denotes the optimal choice for c. The sixth column denotes the competitive ratio achieved by using FairShare with the corresponding cost sharing method and choice of c (we assume we know $\Theta(N)$ and use Corrollary 2). Note that the results for facility location and economic lot sizing assuming each customer requests one unit, we discuss extensions in Section 6. The final column shows lower bounds on the best possible competitive ratios, shown in Section 5 and the Appendix.

Problem	α	β	γ	c	FairShare	Lower Bound
Facility Location [37]	$\log N$ [40]	3	1	$\sqrt{\log N}$	$O(\sqrt{\log N})$	$\Omega\left(\sqrt{\frac{\log N}{\log \log N}}\right)$
Economic Lot Sizing [48]	$\log N$ [40]	$\max(\frac{b}{h}, \frac{h}{b})$	1	$\sqrt{\log N}$	$O(\sqrt{\log N})$	$\Omega\left(\sqrt{\frac{\log N}{\log \log N}}\right)$
Steiner Tree [27]	$\Theta(\log^2 N)$ [39]	2	1	$\log N$	$O(\log N)$	$\Omega(\sqrt{\log N})$
Steiner Forest [29]	$\Theta(\log^2 N)$ [15]	2	1	$\log N$	$O(\log N)$	$\Omega(\sqrt{\log N})$
SSROB [22]	$\Theta(\log^2 N)$ [40]	4.6	1	$\log N$	$O(\log N)$	$\Omega(\sqrt{\log N})$
MROB [40]	$\Theta(\log^2 N)$ [40]	$O(1)$	1	$\log N$	$O(\log N)$	$\Omega(\sqrt{\log N})$

Based on the results of Brenner and Schäfer [10] and Bleischwitz and Schoppmann [9], we can also get $O(\sqrt{\log N})$ competitive ratios with FairShare for the machines scheduling problems with online customer selection using identical and related machines, respectively. Finally, it is worth noting that using the cost sharing method of Moulin and Shenker [35] we can get $O(\sqrt{\log N})$ competitive ratios for submodular problems with online customer selection, although this does not match the optimal guarantee of 2 in Elmachtoub and Levi [16]. To the best of our knowledge, the results in the last two columns are the first of their kind with one exception in that Qian and Williamson [38] found a similar guarantee for the Steiner tree problem with online customer selection.

5. Lower Bound for Facility Location Problem with Online Customer Selection   In this section, we provide lower bounds on the competitive ratio for any algorithm, deterministic or randomized, for the facility location problems with online customer selection. The proof for the economic lot sizing problem is almost identical and therefore omitted. In the Appendix, we provide the proof for the Steiner tree problem which has many similar features to the proof below. The technique relies on constructing an instance similar to the one used in the lower bound proof for the online facility location problem without customer selection in Fotakis [17]. However, we modify the demand instance and add rejection costs in order to obtain a worst case adversary.

THEOREM 2. *The competitive ratio for any deterministic or randomized algorithm for the facility location problem with online customer selection is* $\Omega\left(\sqrt{\frac{\log N}{\log\log N}}\right)$.

Proof.   To show this lower bound, we need to specify an instance of the problem such that no algorithm can come with in a factor of $\Omega\left(\sqrt{\frac{\log N}{\log\log N}}\right)$ of the optimal offline cost, where N is the total number of customers. The proof is by construction, and we start by building a hierarchically well-separated binary tree (Bartal [6]) of depth h (Figure 1), where h will be specified later. The root node will be level 0, its children will be level 1, and so on. There is a potential facility at each leaf in the tree, each with a facility cost f. The distance from a level-i node to its children will be D/m^i, where D and m will also be specified later. For a node v in the tree, we define T_v to be the subtree rooted at v. We now state two easily verifiable facts about the tree.

1. Let v be a level-i node. Then the distance from v to a node not in T_v is at least D/m^{i-1}.

2. Let v be a level-i node. Then the distance from v to any node in T_v is at most $D/(m^{i-1}(m-1))$.

We will now provide a distribution of customer sequences and give a lower bound of $\Omega\left(\sqrt{\frac{\log N}{\log\log N}}\right)$ on the competitive ratio of any deterministic algorithm. Using Yao's principle (Yao [49]), this achieves the desired result. The distribution can be described according to the following procedure. See Figure 1 for an example. The sequence $v_1,\ldots,v_h$ denotes the locations of where the customers' locations will occur, and v is simply keeping track of how to move along the tree. We will assign each v_i to either $V^{\hat{\mathcal{A}}}$ or $V^{\hat{\mathcal{R}}}$ depending on whether we will accept or reject it in the feasible offline
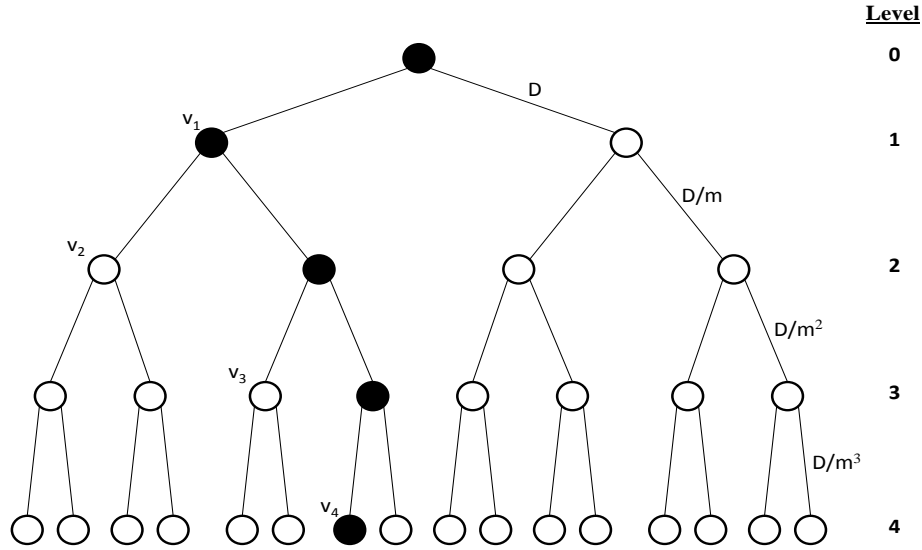
FIGURE 1. Example of the tree metric and a customer sequence.

Note. In this figure, each circle represents a node, and the lines represent where the node can travel to. The black nodes denote a random path that the v node took in the demand sequence construction. In this example, $V^{\hat{A}} = \{v_1, v_4\}$ and $V^{\hat{R}} = \{v_2, v_3\}$.

solution we will consider.

---

1. Let $v := $root, $i := 0$, $V^{\hat{\mathcal{A}}} = \emptyset$ and $V^{\hat{\mathcal{R}}} = \emptyset$.

2. Let $i = i + 1$. Set $v_i := $left child of v. Generate m^{i-1} consecutive demands at v_i each with rejection cost of $r_i = D/(m^{i-1}\sqrt{m})$.

3. With probability $1 - 1/\sqrt{m}$, set $v := $left child of v and let $V^{\hat{\mathcal{A}}} = \{v_i\} \cup V^{\hat{\mathcal{A}}}$. Otherwise, with probability $1/\sqrt{m}$, set $v := $right child of v and $V^{\hat{\mathcal{R}}} = \{v_i\} \cup V^{\hat{\mathcal{R}}}$.

4. If $i < h$, go to Step 2. Else, Stop.

We now compute an upper bound on the expected cost of the optimal offline solution, $\mathbb{E}[\text{OPT}]$, by constructing a feasible solution for any sample path. Let $\hat{\mathcal{A}}$ denote all the customers located at a node in $V^{\hat{\mathcal{A}}}$ and let $\hat{\mathcal{R}}$ denote all the customers located at a node in $V^{\hat{\mathcal{R}}}$. In our feasible solution, we will accept $\hat{\mathcal{A}}$ and reject $\hat{\mathcal{R}}$. Note that there is a common facility in the subtrees of all the nodes that have customers in $\hat{A}$, which is where how they will be served in this feasible solution. We call this facility $\hat{F}$. We can then show that

$$\mathbb{E}[\text{OPT}] \le \mathbb{E}[P(\hat{A}) + R(\hat{R})]$$

$$\le f + \frac{D}{m-1}\mathbb{E}[|V^{\hat{A}}|] + R(\hat{R})$$

$$\le f + \frac{D}{m-1}\mathbb{E}[|V^{\hat{A}}|] + \frac{D}{\sqrt{m}}\mathbb{E}[|V^{\hat{R}}|]$$

$$= f + \frac{D}{m-1}\frac{h\sqrt{m}-h}{\sqrt{m}} + \frac{D}{\sqrt{m}}\frac{h}{\sqrt{m}}$$

$$\le f + 2\frac{hD}{m-1}$$

The first inequality follows from optimality. From Fact 2, it follows that for each location $v_i \in V^{\hat{A}}$, the service cost for all the customers at v_i to $\hat{F}$ is $m^{i-1}\frac{D}{m^{i-1}(m-1)} = \frac{D}{m-1}$, which implies the second inequality. For each location $v_i \in V^{\hat{R}}$, the rejection cost for all the customers at v_i will be $m^{i-1}\frac{D}{m^{i-1}\sqrt{m}} = \frac{D}{\sqrt{m}}$, which implies the third inequality. The fourth line follows from Step 3 of the customer sequence construction and the fact that there are h levels in total(probability of $v_i \in V^{\hat{A}}$ is $1 - 1/\sqrt{m}$ for all i). The last inequality follows from simple algebra.

We now focus on the cost of an arbitrary deterministic algorithm, ALG, on this input. Let us again refer to $\mathcal{A}$ and $\mathcal{R}$ as the accepted and rejected customers by ALG. Let $V^{\mathcal{A}} \subseteq \{v_1, \ldots, v_h\}$ be the locations where at least half of the customers are in $\mathcal{A}$ and let $V^{\mathcal{R}}$ be the remaining locations. The expected cost of ALG is then

$$\mathbb{E}[\text{ALG}] = \mathbb{E}[P(\mathcal{A}) + R(\mathcal{R})]$$

$$\ge \mathbb{E}[P(\mathcal{A} \cap \hat{\mathcal{R}})] + \mathbb{E}[R(\mathcal{R} \cap \hat{\mathcal{A}})]$$

$$\ge \min\{f, D/2\}\mathbb{E}[|V^{\mathcal{A}} \cap V^{\hat{\mathcal{R}}}|] + \mathbb{E}[R(\mathcal{R} \cap \hat{\mathcal{A}})]$$

$$\ge \min\{f, D/2\}\mathbb{E}[|V^{\mathcal{A}} \cap V^{\hat{\mathcal{R}}}|] + \frac{D}{2\sqrt{m}}\mathbb{E}[|V^{\mathcal{R}} \cap V^{\hat{\mathcal{A}}}|]$$

$$\ge \min\{f, D/2\}\frac{\mathbb{E}[|V^{\mathcal{A}}|]}{\sqrt{m}} + \frac{D}{2\sqrt{m}}\frac{\sqrt{m}\mathbb{E}[|V^{\mathcal{R}}|] - \mathbb{E}[|V^{\mathcal{R}}|]}{\sqrt{m}}$$

$$\ge \min\{f, D/4\}\frac{\mathbb{E}[|V^{\mathcal{A}}|]}{\sqrt{m}} + \frac{D}{4\sqrt{m}}E[|V^{\mathcal{R}}|]$$

The first inequality follows from monotonicity of $P(\cdot)$ and $R(\cdot)$. Note that to serve the customers

in $\mathcal{A}$, we need to serve at least half of the customers at each location $v_i \in V^{\mathcal{A}} \cap V^{\hat{\mathcal{R}}}$. By construction

of $V^{\hat{R}}$, none of the subtrees of the nodes in $V^A \cap V^{\hat{R}}$ intersect. Thus, for each v_i we must either

construct a facility in T_{v_i} or serve the customers from another facility which, by Fact 1, will cost

$\frac{m^{i-1}}{2} \frac{D}{m^{i-1}} = \frac{D}{2}$. Thus, the second inequality follows from the lower bound just derived of serving

the accepted customers at $V^A \cap V^{\hat{R}}$. The third inequality follows from the fact that we need

to reject at least half of the customers located at $V^R \cap V^{\hat{A}}$. For a given location, this will cost

$\frac{m^{i-1}}{2} \frac{D}{m^{i-1}\sqrt{m}} = \frac{D}{2\sqrt{m}}$. The fourth inequality follows from the fact that the probability that any node

v_i is in $V^{\hat{A}}$ is $1 - 1/\sqrt{m}$ by Step 3 of the construction, regardless of the fact that $v_i \in V^R$ as well.

The last inequality follows when $m \geq 4$.

Now we will carefully set $h := m$ and $D := 4f$. Then the competitive ratio of ALG on this instance

will be

$$
\begin{aligned}
\frac{\mathbb{E}[\text{ALG}]}{\mathbb{E}[\text{OPT}]} &\geq \frac{\min\{f, D/4\}\frac{\mathbb{E}[|V^A|]}{\sqrt{m}} + \frac{D}{4\sqrt{m}}E[|V^R|]}{f + 2\frac{hD}{m-1}} \\
&= \frac{\frac{\mathbb{E}[|V^A|]}{\sqrt{m}} + \frac{1}{\sqrt{m}}E[|V^R|]}{1 + 8\frac{m}{m-1}} \\
&= \frac{\sqrt{m}}{1 + 8\frac{m}{m-1}} \\
&= \Omega(\sqrt{m})
\end{aligned}
$$

The first inequality follows directly from the two bounds we derived above. The first equality

follows from canceling out f terms. The second equality follows from the fact that the expected

total number of locations is m by construction.

Finally, we need to determine m. Since there are $1 + m + \ldots + m^{m-1} \approx m^m$ total units of demand,

then the most m can be is $\Theta(\frac{\log N}{\log \log N})$. Thus, the competitive ratio for any algorithm is at least

$\Omega\left(\sqrt{\frac{\log N}{\log \log N}}\right)$. $\quad \square$

6. Extensions for Online Market Selection   In this section, we consider a more typical scenario where each customer actually has a multi-unit request that can even be across multiple time periods or locations. We refer to this type of customer as a market, which requires special consideration since cost sharing schemes have typically only been developed with single demand customers in mind. Geunes et al. [18] gives a 2.06-approximation algorithm for several problems with *offline* market selection. In order to handle *online market selection*, we simply create a new cost sharing scheme that is based on summing up the cost shares for each demand when the original cost sharing scheme is used.

For example, assume we have a cross-monotonic, (β, γ)-budget balanced, α-summable cost sharing scheme $\hat{\chi}$. However, $\hat{\chi}$ assumes that each player has one demand request. If customers have multiple demand requests, which create markets, and they only want to be served if their entire market of demands can be served, then we need to come up with a new cost sharing scheme. Specifically, for each customer j, let T_j be the set of demands that customer j wants. We will define the new cost sharing scheme χ as $\chi(j, \cdot) = \sum_{k \in T_j} \hat{\chi}(k, \cdot)$. Clearly χ maintains the cross-monotonicity and (β, γ)-budget balanced properties. It remains α summable, but the input to $\alpha(\cdot)$ is now the total number of demands, rather than the number of customers. Thus, the competitive ratios achieved by FairShare are now dependent on the total number of units of demand requested. In addition, our lower bounds also have the feature that the lower bound will also depend on the the total number of units of demand request. Specifically, all our results are trivially extended by just replacing N by the total number of demands requested.

7. Conclusion   In this paper we considered a general class of online customer selection problems where customers arrive sequentially and each customer must be either accepted or rejected upon arrival, without knowledge of the future remaining customer sequence. The goal is to minimize the production cost for the accepted customers plus the rejection cost of the rejected customers. We provided a general algorithm called FairShare for this class of problems that is based on the use of cost sharing mechanisms. These cost sharing mechanisms aim to naturally distribute the

cost of production to the "players" in the cooperative game in such a way that the allocation is fair and recovers most of the cost. Therefore, the cost shares very closely approximate the cost incurred by each player, and we are able to leverage this information to determine whether to accept or reject customers. Our algorithm provides poly-logarithmic competitive ratio guarantees for a variety of facility location, network design, and inventory management problems with online customer selection. We also showed that these guarantees are close to best possible by constructing customer sequences that result in poly-logarithmic competitive ratios for all algorithms. Future research directions would be to bridge the gap between the upper and lower bounds we developed and to consider further applications of mechanism design in online optimization.

Appendix. Lower Bound for Steiner Tree with Online Customer Selection

In this section, we derive a lower bound on the competitive ratio for the Steiner tree problem with online customer selection. The proof is similar in spirit to Imase and Waxman [23] who provided a lower bound for the online Steiner problem without customer selection. Note that this proof also holds for the Steiner forest, single-commidity rent-or-buy problem, and multi-commodity rent-or-buy problems with online customer selection since they are all generalizations.

THEOREM 3. *The competitive ratio for any deterministic or randomized algorithm for the Steiner tree problem with online customer selection is* $\Omega\left(\sqrt{\log N}\right)$.

Proof. We begin by constructing the same graph as in Imase and Waxman [23]. The procedure to build graph G_k is as follows. Begin with two nodes, v_0 (bottom) and v_1 (top) connected by an edge of length 1, and call this graph G_0. Now to create graph G_k, take every edge in G_{k-1} and replace it by 2 paths, each consists of 2 new edges and 1 new node. The lengths of the 4 new edges will each be half of the length of the original edge, so their length is 2^k. The new nodes are called level k nodes, and each pair of new nodes created are called k-adjacent to each other. See Figure 2, 3, 4, and 5 for an example of G_3 (nodes are labeled according to their level, each edge is of length 1/8).

The procedure below demonstrates how we will generate customers. We begin by having 1 customer at level 1, 2 customers at level 2, 4 customers at level 3, and so on until we have 2^{k-1} customers at level k. The rejection cost of a level i customer is $r_i = 1/(2^{i-1}\sqrt{k})$. $V^{\hat{\mathcal{A}}}$ and $V^{\hat{\mathcal{R}}}$ denote the node levels that are accepted and rejected, respectively, by the feasible solution we will consider. See Figures 2-5 for examples on G_3 of the customer sequence.
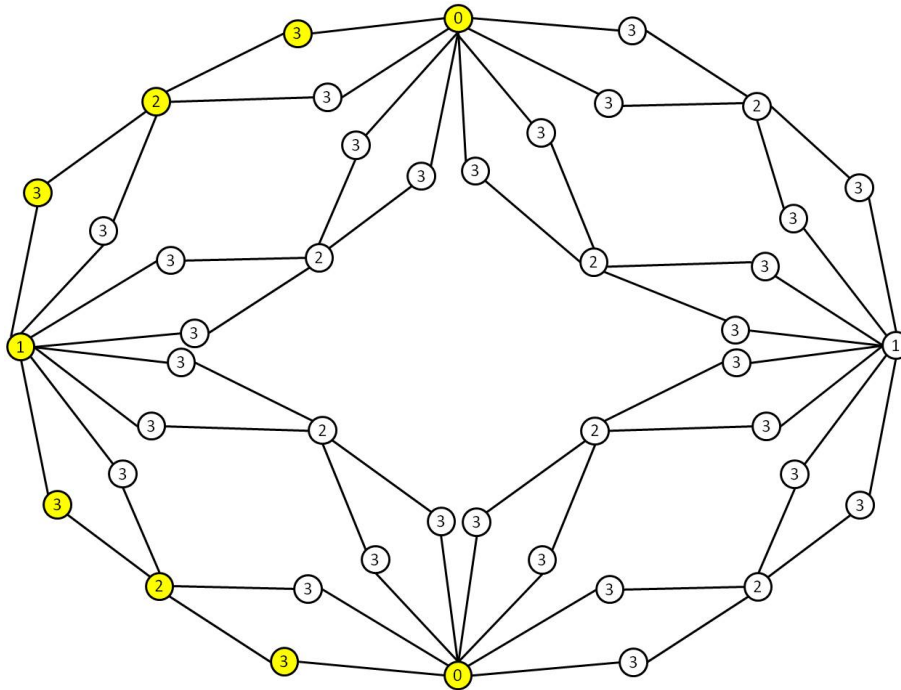
FIGURE 2. Example of the Steiner tree problem with online customer selection.

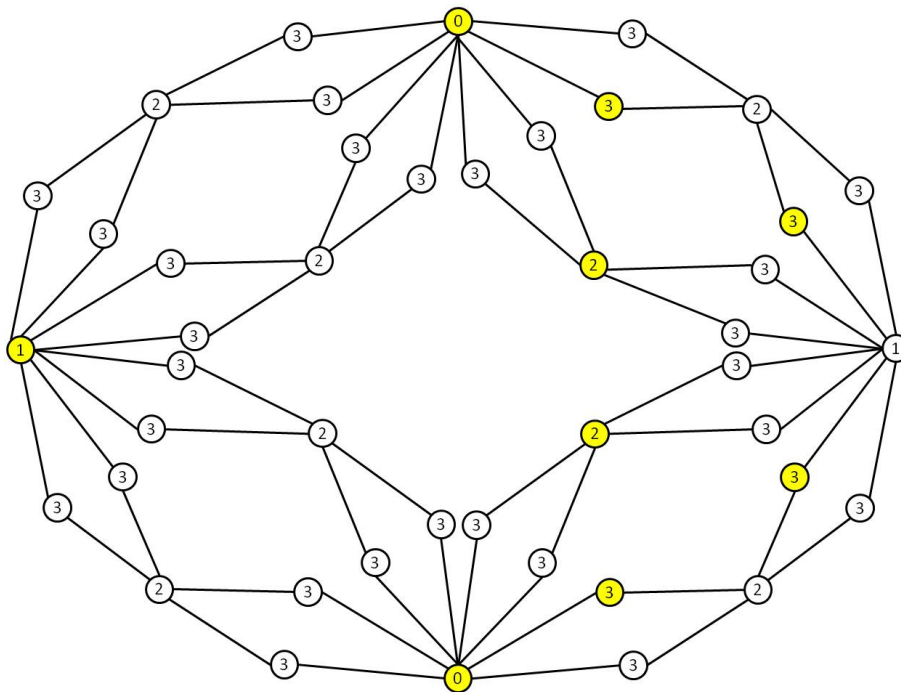*Note.* In this figure, $V^{\hat{\mathcal{A}}} = \{0,1,2,3\}$.

FIGURE 3. Example of the Steiner tree problem with online customer selection.

Note. In this figure, $V^{\hat{\mathcal{A}}} = \{0\}$.
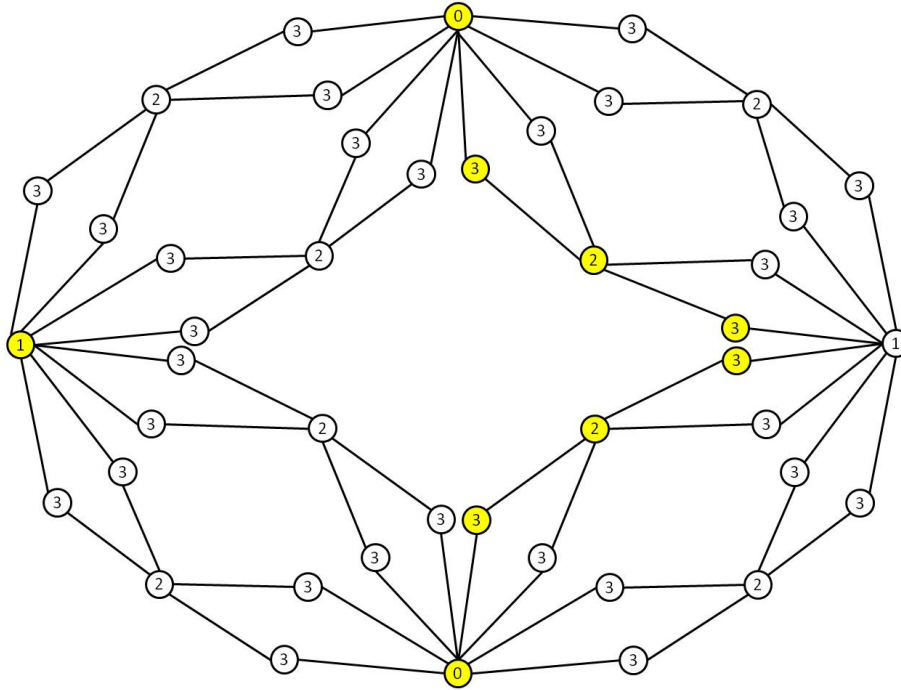
FIGURE 4. Example of the Steiner tree problem with online customer selection.

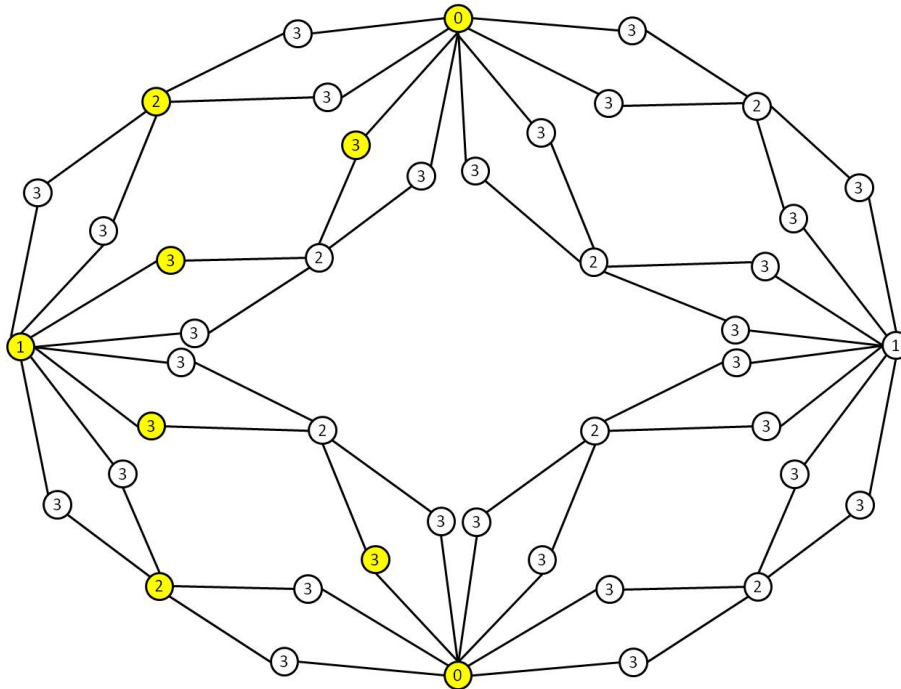*Note.* In this figure, $V^{\hat{\mathcal{A}}} = \{0, 2, 3\}$.

FIGURE 5. Example of the Steiner tree problem with online customer selection.

Note. In this figure, $V^{\hat{\mathcal{A}}} = \{0, 1, 3\}$.

1. Let $V := v_0$, $i := 0$, $V^{\hat{\mathcal{A}}} = \{0\}$ and $V^{\hat{\mathcal{R}}} = \emptyset$. Let the first two customers be v_0 and v_1, each with infinite rejection cost (must accept).

2. Let $i = i + 1$. Set $V_i :=$ left i-level nodes of V. Generate 2^{i-1} customers, one at each node in V_i, each with rejection cost of $r_i = 1/(2^{i-1}\sqrt{k})$.

3. With probability $1 - 1/\sqrt{k}$, set $V := V_i$ and let $V^{\hat{\mathcal{A}}} = \{i\} \cup V^{\hat{\mathcal{A}}}$. Otherwise, with probability $1/\sqrt{k}$, set $V :=$ i-adjacent nodes of V_i and $V^{\hat{\mathcal{R}}} = \{i\} \cup V^{\hat{\mathcal{R}}}$.

4. If $i < k$, go to Step 2. Else, Stop.

By construction, it's clear that all the customer nodes with levels in $V^{\hat{\mathcal{A}}}$ can be connected by a single path of length 1. The cost of rejecting all the nodes at any given level is $2^{i-1}r_i = 2^{i-1}(1/(2^{i-1}\sqrt{k})) = 1/\sqrt{k}$. The expected number of levels that are rejected is $\mathbb{E}[|V^{\hat{\mathcal{R}}}|]$, which we can compute to be $(1/\sqrt{k})k = \sqrt{k}$ since the probability that any level is in $V^{\hat{\mathcal{R}}}$ is $1/\sqrt{k}$. The total cost of the feasible solution implied by $\hat{\mathcal{A}}$ and $\hat{\mathcal{R}}$ is then $1 + \mathbb{E}[|V^{\hat{\mathcal{R}}}|]/\sqrt{k} = 2$.

Now let $V^{\mathcal{A}}$ be the levels where an online algorithm accepted at least half of the customers, and let $V^{\mathcal{R}}$ be the levels where an online algorithm rejected at least half of the customers. By construction, in expectation $1/\sqrt{k}$ of the levels in $V^{\mathcal{A}}$ will be in $V^{\hat{\mathcal{R}}}$ due to the random customer distribution. By construction, nodes in level $i \in V^{\hat{\mathcal{R}}}$ can only be connected to any other nodes if they each pay a cost of at least $1/2^i$. Therefore, the cost of connecting half the nodes in any level of $V^{\hat{\mathcal{R}}}$ is $(1/2)(2^{i-1})(1/2^i) = 1/4$. Since on average $1/\sqrt{k}$ of levels in $V^{\mathcal{A}}$ are in in $V^{\hat{\mathcal{R}}}$, then the cost of serving nodes in levels of $V^{\mathcal{A}}$ is at least $\mathbb{E}[|V^{\mathcal{A}}|]/4\sqrt{k}$. The cost of rejecting half of the nodes in levels of $|V^{\mathcal{R}}|$ is computed similarly as before, and is $\mathbb{E}[|V^{\mathcal{R}}|]/2\sqrt{k}$. Therefore the total cost of the online algorithm is at least $\mathbb{E}[|V^{\mathcal{A}}|]/4\sqrt{k} + \mathbb{E}[|V^{\mathcal{R}}|]/4\sqrt{k} = \sqrt{k}/4$.

Thus, the cost ratio between the online and feasible solution is $O(\sqrt{k})$. So if we have $2 + 2^0 + 2^1 + \ldots + 2^k = N$ total customers, then k is at most $\Theta(\log N)$ which completes the result. $\square$

References

[1] Albers, Susanne. 2003. Online algorithms: a survey. *Mathematical Programming* **97**(1-2) 3–26.

[2] Archer, A., M.H. Bateni, M.T. Hajiaghayi, H. Karloff. 2009. Improved approximation algorithms for prize-collecting steiner tree and tsp. *Foundations of Computer Science, 2009. FOCS'09. 50th Annual IEEE Symposium on*. IEEE, 427–436.

[3] Axsater, S. 1982. Worst case performance for lot sizing heuristics. *European Journal of Operational Research* **9**(4) 339–343.

[4] Balas, E. 1989. The prize collecting traveling salesman problem. *Networks* **19**(6) 621–636.

[5] Ball, M.O., M. Queyranne. 2009. Toward robust revenue management: Competitive analysis of online booking. *Oper. Res.* **57**(4) 950–963.

[6] Bartal, Y. 1996. Probabilistic approximation of metric spaces and its algorithmic applications. *Foundations of Computer Science, 1996. Proceedings., 37th Annual Symposium on*. IEEE, 184–193.

[7] Bhaskaran, Sreekumar R, Vish Krishnan. 2009. Effort, revenue, and cost sharing mechanisms for collaborative new product development. *Management Science* **55**(7) 1152–1169.

[8] Bienstock, D., M.X. Goemans, D. Simchi-Levi, D. Williamson. 1993. A note on the prize collecting traveling salesman problem. *Mathematical Programming* **59**(1) 413–420.

[9] Bleischwitz, Y., F. Schoppmann. 2008. New efficiency results for makespan cost sharing. *Information Processing Letters* **107**(2) 64–70.

[10] Brenner, J., G. Schäfer. 2007. Cost sharing methods for makespan and completion time scheduling. *STACS 2007* 670–681.

[11] Buchbinder, N., T. Kimbrelt, R. Levi, K. Makarychev, M. Sviridenko. 2008. Online make-to-order joint replenishment model: primal dual competitive algorithms. *Proceedings of the nineteenth annual ACM-SIAM symposium on Discrete algorithms*. Society for Industrial and Applied Mathematics, 952–961.

[12] Byrka, J., F. Grandoni, T. Rothvoß, L. Sanità. 2010. An improved lp-based approximation for steiner tree. *Proceedings of the 42nd ACM symposium on Theory of computing*. ACM, 583–592.

[13] Cachon, G.P. 2003. Supply chain coordination with contracts. *Handbooks in operations research and management science* **11** 229–340.

[14] Charikar, M., S. Khuller, D.M. Mount, G. Narasimhan. 2001. Algorithms for facility location problems with outliers. *Proceedings of the twelfth annual ACM-SIAM symposium on Discrete algorithms*. Society for Industrial and Applied Mathematics, 642–651.

[15] Chawla, S., T. Roughgarden, M. Sundararajan. 2006. Optimal cost-sharing mechanisms for steiner forest problems. *Internet and Network Economics* 112–123.

[16] Elmachtoub, A.N., R. Levi. 2013. Supply chain management with online customer selection. *Unpublished Manuscript* .

[17] Fotakis, D. 2008. On the competitive ratio for online facility location. *Algorithmica* **50**(1) 1–57.

[18] Geunes, J., R. Levi, H.E. Romeijn, D.B. Shmoys. 2009. Approximation algorithms for supply chain planning and logistics problems with market choice. *Mathematical Programming* 1–22.

[19] Geunes, J., H.E. Romeijn, K. Taaffe. 2006. Requirements planning with pricing and order selection flexibility. *Oper. Res.* **54**(2) 394.

[20] Gilbert, EN, HO Pollak. 1968. Steiner minimal trees. *SIAM Journal on Applied Mathematics* 1–29.

[21] Goemans, M.X., D.P. Williamson. 1995. A general approximation technique for constrained forest problems. *SIAM Journal on Computing* **24** 296.

[22] Gupta, A., A. Srinivasan, É. Tardos. 2004. Cost-sharing mechanisms for network design. *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques* 139–150.

[23] Imase, M., B.M. Waxman. 1991. Dynamic steiner tree problem. *SIAM Journal on Discrete Mathematics* **4** 369.

[24] Jaillet, Patrick, Xin Lu. 2013. Online traveling salesman problems with rejection options. Tech. rep., Working paper, Operations Research Center, Massachusetts Institute of Technology.

[25] Jain, K. 2001. A factor 2 approximation algorithm for the generalized steiner network problem. *Combinatorica* **21**(1) 39–60.

[26] Jain, K., M. Mahdian. 2007. Cost sharing. *Algorithmic game theory* 385–410.

[27] Jain, K., V.V. Vazirani. 2002. Equitable cost allocations via primal-dual-type algorithms. *Proceedings of the thiry-fourth annual ACM symposium on Theory of computing*. ACM, 313–321.

[28] Kim, Sang-Hyun, Serguei Netessine. 2013. Collaborative cost reduction and component procurement under information asymmetry. *Management Science* **59**(1) 189–206.

[29] Könemann, J., S. Leonardi, G. Schäfer. 2005. A group-strategyproof mechanism for steiner forests. *Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms*. Society for Industrial and Applied Mathematics, 612–619.

[30] Kumar, A., A. Gupta, T. Roughgarden. 2002. A constant-factor approximation algorithm for the multicommodity rent-or-buy problem. *Foundations of Computer Science, 2002. Proceedings. The 43rd Annual IEEE Symposium on*. IEEE, 333–342.

[31] Li, Shi. 2012. A 1.488 approximation algorithm for the uncapacitated facility location problem. *Information and Computation* .

[32] Li, Yu, Xiu N Du D, D Xu. 2012. *Improved approximation algorithms for the facility location problems with linear/submodular penalty*. Faculty of Business Administration, University of New Brunswick.

[33] Meyerson, A. 2002. Online facility location. *Proceedings of 42nd IEEE Symposium onFoundations of Computer Science*. IEEE, 426–431.

[34] Moulin, H. 1999. Incremental cost sharing: Characterization by coalition strategy-proofness. *Social Choice and Welfare* **16**(2) 279–320.

[35] Moulin, H., S. Shenker. 2001. Strategyproof sharing of submodular costs: budget balance versus efficiency. *Economic Theory* **18**(3) 511–533.

[36] Nagarajan, M., G. Sosic. 2008. Game-theoretic analysis of cooperation among supply chain agents: Review and extensions. *European Journal of Operational Research* **187**(3) 719–745.

[37] Pál, M., É. Tardos. 2003. Group strategy proof mechanisms via primal-dual algorithms. *Foundations of Computer Science, 2003. Proceedings. 44th Annual IEEE Symposium on*. IEEE, 584–593.

[38] Qian, J., D. Williamson. 2011. An o (logn)-competitive algorithm for online constrained forest problems. *Automata, Languages and Programming* 37–48.

[39] Roughgarden, T., M. Sundararajan. 2006. New trade-offs in cost-sharing mechanisms. *Proceedings of the thirty-eighth annual ACM symposium on Theory of computing*. ACM, 79–88.

[40] Roughgarden, T., M. Sundararajan. 2007. Optimal efficiency guarantees for network design mechanisms. *Integer Programming and Combinatorial Optimization* 469–483.

[41] Shmoys, D.B., É. Tardos, K. Aardal. 1997. Approximation algorithms for facility location problems. *Proceedings of the twenty-ninth annual ACM Symposium on Theory of Computing*. ACM, 265–274.

[42] Slotnick, Susan A. 2011. Order acceptance and scheduling: a taxonomy and review. *European Journal of Operational Research* **212**(1) 1–11.

[43] Talwar, K. 2006. The single-sink buy-at-bulk lp has constant integrality gap. *Integer Programming and Combinatorial Optimization* 475–486.

[44] van den Heuvel, W., O.E. Kundakcioglu, J. Geunes, H.E. Romeijn, T.C. Sharkey, A.P.M. Wagelmans. 2009. Integrated Market Selection and Production Planning: Complexity and Solution Approaches. *Unpublished Manuscript* .

[45] van den Heuvel, W., APM Wagelmans. 2009. Worst case analysis for a general class of on-line lot-sizing heuristics. *Oper. Res.* .

[46] Wagner, H.M., T.M. Whitin. 1958. Dynamic Version of the Economic Lot Size Model. *Management Science* **5**(1) 89–96.

[47] Wagner, M.R. 2010. Fully Distribution-Free Profit Maximization: The Inventory Management Case. *Math. Oper. Res.* **35**(4) 728.

[48] Xu, D., R. Yang. 2009. A cost-sharing method for an economic lot-sizing game. *Operations Research Letters* **37**(2) 107–110.

[49] Yao, A. C. 1977. Probabilistic computation: towards a unified measure of complexity. *IEEE Symposium on Foundations of Computer Science*. 222–227.