Primal-Dual Algorithms for Deterministic Inventory Problems

Retsef Levi* Robin Roundy† David B. Shmoys‡

Abstract

We consider several classical models in deterministic inventory theory: the single-item lot-sizing problem, the joint replenishment problem, and the multi-stage assembly problem. These inventory models have been studied extensively, and play a fundamental role in broader planning issues, such as the management of supply chains. For each of these problems, we wish to balance the cost of maintaining surplus inventory for future demand against the cost of replenishing inventory more frequently. For example, in the joint replenishment problem, demand for several commodities is specified over a discrete finite planning horizon, the cost of maintaining inventory is linear in the number of units held, but the cost incurred for ordering a commodity is independent of the size of the order; furthermore, there is an additional fixed cost incurred each time a non-empty subset of commodities is ordered. The goal is to find a policy that satisfies all demands on time and minimizes the overall holding and ordering cost.

We shall give a novel primal-dual framework for designing algorithms for these models that significantly improve known results in several ways: the performance guarantees for the quality of the solutions improve on or match previously known results; the performance guarantees hold under much more general assumptions about the structure of the costs, and the algorithms and their analysis are significantly simpler than previous known results. Finally, our primal-dual framework departs from the structure of previously studied primal-dual approximation algorithms in significant ways, and we believe that our approach may find application in other settings. More specifically, we provide 2-approximation algorithms to the joint replenishment problem and to the assembly problem, and solve the single-item lot-sizing problem to optimality. The results for the joint replenishment and the lot-sizing problems also hold for their generalizations with back orders allowed. As a by product of our work, we prove known and new upper bounds on the integrality gap of some LP relaxations of the above mentioned problems.

*rl227@cornell.edu. School of ORIE, Cornell University, Ithaca, NY 14853. Research supported partially by a grant from Motorola and NSF grant CCR-9912422.

†robin@orie.cornell.edu. School of ORIE, Cornell University, Ithaca, NY 14853. Research supported partially by a grant from Motorola, NSF grant DMI-0075627, and the Querétaro Campus of the Instituto Tecnológico y de Estudios Superiores de Monterrey.

‡shmoys@cs.cornell.edu. School of ORIE and Dept. of Computer Science, Cornell University, Ithaca, NY 14853. Research supported partially by NSF grant CCR-9912422.

1 Introduction

In this paper, we consider several classical models in deterministic inventory theory: the single-item lot-sizing problem, the joint replenishment problem (JRP) and the multi-stage assembly problem. These inventory models have been studied extensively over the years, in a number of different settings, and play a fundamental role in broader planning issues, such as the management of supply chains (see, e.g., [3, 13]). We shall consider the variants in which there is a discrete notion of time with a finite planning horizon, and the demand is deterministic (known in advance) but dynamic, i.e., it varies over the planning horizon.

Each of the inventory models that we consider has the following characteristics. There are N commodities (or equivalently, items) that are needed over a planning horizon consisting of T time periods; for each time period and each commodity, there is a demand for a specified number of units of that commodity. To satisfy these demands, an order may be placed in each time period. For each commodity i ordered, a *fixed ordering cost* K_i is incurred, which is independent of the number of units ordered from that commodity. The order placed in time period t may be used to satisfy demand in time period t or any subsequent point in time. In addition, the demand in time period t must be satisfied completely by orders that have been placed no later than time period t. (In the inventory literature, these assumptions are usually referred to as "neither back orders nor lost sales are allowed".) Since the cost of ordering a commodity is independent of the number of units ordered, there is an incentive to place large orders, to meet the demand not just for the current time period, but for subsequent time periods as well. This is balanced by a cost incurred for holding inventory over time periods. We will let h^i_{st} denote this *holding cost*, that is, the cost incurred by ordering one unit of inventory in period s, and using it to meet the demand for item i in period t. We will assume that h^i_{st} is non-negative and, for each (i, t), is a non-increasing function of s. (Note that in particular, we do not require subadditivity; we could have that $h^i_{rt} > h^i_{rs} + h^i_{st}$ for some $r < s < t$.) The goal is to find a policy of orders that satisfies all demands on time and minimizes the overall holding and ordering cost.

The details of the three inventory models are as follows. In the *single-item lot-sizing problem*, we have a single item ($N = 1$) with specified demands over T time periods $(d_1, .., d_T)$. In the *joint replenishment problem* we have N commodities, where for each commodity $i = 1, \ldots, N$, and for each time period $t = 1, \ldots, T$, there is a specified non-negative demand d_{it}. In addition to the item ordering costs, K_i, $i = 1, \ldots, N$, *any* order incurs what we call *a joint ordering cost* K_0, independent of the (nonempty) subset of commodities that are included in the order (and again, independent of the (positive) number of units for each commodity included). The joint ordering cost creates a dependency between the different commodities and complicates the structure of the optimal policy. The holding cost follows the same structure described above.

In the *assembly problem*, we have a somewhat more involved structure. As part of the input, we also specify a rooted directed in-tree, where each node in the tree corresponds to an item, and we assume that the items are indexed so that $i > j$ for each edge (i, j) in the tree. Node (or item) 1, the root of the tree, is facing external demands over T time periods $(d_1, .., d_T)$. A unit of item i is assembled from one unit of each of its predecessor items in the tree. Thus, any unit of item 1 consists of one unit of each of the other items. We again have an ordering cost and holding cost for each item.

We note that the way we model the holding cost is much more general than the most common setting, in which each item i has a linear holding cost, so that the cost of holding one unit from time period s to time period t is equal to $(t - s)h_i$, for some choice of $h_i > 0$ (or to $\sum_{l=s}^{t} h^i_l$ in the more general case). By allowing the more general structure described above, we can capture other important phenomena, such as perishable goods, where the cost of holding an item longer than a specified interval is essentially infinite. The strength of the general holding cost structure is demonstrated in Section 4.3, where we show how to apply the algorithm to the more general JRP model with backorders. As for the ordering cost, we note that our algorithms are applicable also in the presence of time dependent cost parameters as will be specified later on. Furthermore, in addition to the (fixed) ordering cost that is independent of the order size, one can

incorporate a per-unit ordering cost into the holding cost term (as long as we preserve the monotonicity).

In this paper, we describe a unified novel primal-dual algorithmic framework that provides optimal and near-optimal solutions to the three inventory models described above. Our main result is a 2-approximation algorithm for the joint replenishment problem. By this we mean that for any instance of the problem, our algorithm computes a feasible solution in polynomial-time, with cost that is guaranteed to be no more than twice the optimal cost. The joint replenishment problem is NP-hard [2], but it can be solved in polynomial-time by dynamic programming for a fixed number of commodities, or for a fixed number of time periods [28, 26, 16], (by fixing the times at which joint orders are placed the problem decomposes by item). LP-based techniques have not previously played a significant role in the design of approximation algorithms for NP-hard deterministic inventory problems with constant performance guarantee. LP-rounding was applied to a more general problem by Shen, Simchi-Levi, and Teo [23], but this yielded a guarantee of only $O(\log N + \log T)$. This absence of results is particularly surprising in light of the fact that it has long been understood that these problems admit integer programming formulations with strong linear programming relaxations, i.e., that provide tight lower bounds (see, e.g., [14, 19, 20]). These formulations are closely related to formulations that have been studied for the facility location problem, which has also been a source of intense study for approximation algorithms. Our performance guarantee improves significantly on the results of Joneja [15], who only considered the case where all the cost parameters are fixed over time. His paper claims a 3-approximation algorithm for this problem, but it has been pointed out that the proof is flawed [25]. A somewhat different analysis yields a performance guarantee of 5 [18]. Federgrun and Tzur [9] proposed an interesting dynamic programming-based heuristic for the joint replenishment problem, but they assume that cost and demand parameters are bounded by constants.

The single-item lot-sizing problem was shown to be solvable in polynomial time by dynamic programming in the landmark paper of Wagner & Within[27]. Furthermore, Krarup & Bilde [17] showed, in this case, that the facility location-inspired LP has integer optima by means of a primal-dual algorithm, and Bárány. Van Roy, and Wolsey [4] gave yet another proof of this by means of an explicitly generated pair of primal and dual optima (that are computed, ironically, via a dynamic programming computation). Finally, Bertsimas, Teo and Vohra [5] gave a proof, which is based on LP rounding. If we consider our joint replenishment algorithm as applied to the special case of the single-item lot-sizing problem (where, since there is only one item, one can merge the joint ordering cost and the individual item ordering cost into one new ordering cost), then we obtain a new, extremely simple, primal-dual optimization algorithm that also proves the integrality of this LP formulation.

Finally, with some modifications, our primal-dual algorithm can also be applied to the assembly problem to yield a 2-approximation algorithm. Here, we achieve the same approximation ratio as Roundy [21], who gave a 2-approximation algorithm (again for the case where all cost parameters are fixed over time) using a non-linear relaxation and ideas borrowed from continuous-time lot-sizing problems. Although we only match the previous performance guarantee, our approach is much simpler, and it yields the performance guarantee under a much more general cost structure. In particular, under our assumptions on the cost structure, it is easy to show that the assembly problem is NP-hard by a reduction from the joint replenishment problem. However, for the variant of the problem considered by Roundy, it is still not known whether it is NP-hard or not [6].

As a byproduct of our work, we prove upper bounds on the integrality gap of the corresponding LP relaxations, the worst-case ratio between the optimal integer and fractional values; for both the JRP and the assembly problem, we prove an upper bound of 2. In [22], we give a family of instances of the JRP, for which the integrality gap is asymptotically 1.23.

To understand the relationship between these inventory models and facility location problems, one can view placing an order as opening a facility; the demand points that this order serves corresponds to demand points that are served by the open facility. Although these two classes of problems are related, there are also fundamental distinctions between them. For one, the distances implied by this facility location view

of inventory problems is asymmetric and does not satisfy the triangle inequality. For facility location problems, the versions with asymmetric cost metric do not admit constant performance guarantee approximation algorithms (see, e.g., [1, 11, 7]), and so it is particularly interesting that the additional structure in these inventory problems is sufficient to obtain good approximation algorithms. Furthermore, we are interested in multi-commodity models; there has been recent work that considers multi-commodity facility location problems but, of course, with a symmetric cost metric [24].

We note that our algorithms have their intellectual roots in the seminal paper of Jain & Vazirani [12], which gives a primal-dual approximation algorithm for the uncapacitated facility location problem. Nonetheless our algorithms depart from their approach in rather significant ways, as we shall describe in detail in the next section. We believe that this new approach may find applications in other settings.

The rest of the paper is organized in the following way. In Section 2 we describe the generic primal-dual algorithm focusing on the JRP case. Then in Section 3 we first consider the lot-sizing problem as a special case of the JRP and show that the algorithm provides an optimal solution to this special case. In Section 4 we complete the presentation of the algorithm for the JRP case and describe the worst case analysis. We then show how to extend the algorithm for the JRP to the more general case in which back orders are allowed. In Section 5, we describe the modifications in the algorithm and the analysis for the assembly problem. We conclude with some interesting open questions.

2 A primal-dual framework

In this section, we outline the main ideas in our primal-dual framework. We start by giving a high-level description, and then give a more detailed presentation. We shall start by focusing on the JRP. It is straightforward to give an integer programming formulation in which there are 0-1 decision variables that indicate whether the demand for a given commodity in a particular time period is supplied from an order at a specific time period, as well as 0-1 variables that indicate whether an order is placed in a given time period, and whether a particular commodity is included in that order. We shall defer presenting the details of this formulation and the dual of its LP relaxation, since the main ideas of the algorithm can be presented without any explicit reference to the LPs.

Our algorithm works in two phases. In the first phase of the algorithm we simultaneously construct a feasible dual solution and a feasible primal (integer) solution. Each demand point (i, t) has a dual variable b_t^i, which can be interpreted as a budget. In constructing the dual solution, we use a dual-ascent approach. Each budget (i.e., dual variable b_t^i), is initially 0 and is gradually increased until it is frozen at its final value; that is, we never decrease its value.

Unlike the primal-dual algorithm of Jain & Vazirani for the facility location problem (or that of Goemans & Williamson [10] for network design problems), we do not increase the dual variables uniformly. Instead we use a more sophisticated mechanism, which we call a *waveform*. Consider a wave that starts to move from the end of the planning horizon to the beginning (from period T to 1) and let τ be a variable that indicates the current location of the *wavefront*; initially, $\tau = T$. The budget of any unfrozen demand point is then related to the indicator τ. More specifically, each demand point (i, t) keeps its budget fixed at 0 until the wave reaches period t. Moreover, once the wave crosses time t and as long as the budget b_t^i is not frozen, we keep the budget of (i, t) equal to the holding cost of providing d_{it} from τ; that is, $b_t^i = d_{it} \cdot h_{\tau t}^i$, which, for notational convenience, we shall denote $H_{\tau t}^i$ (see Figure 2.1).

Each demand point is going to offer its budget to all potential orders (i.e., time periods) from which it can be served. When offered to some potential order s ($s = 1, \ldots, t$), the budget b_t^i is first used to pay for the holding cost incurred by providing d_{it} from s. The residual budget is then used to pay a share of the item ordering cost K_i with respect to the order s. Once the item ordering cost is completely paid for (by this and other demand points), the residual budget is used to pay a share of the joint ordering cost K_0 with respect to

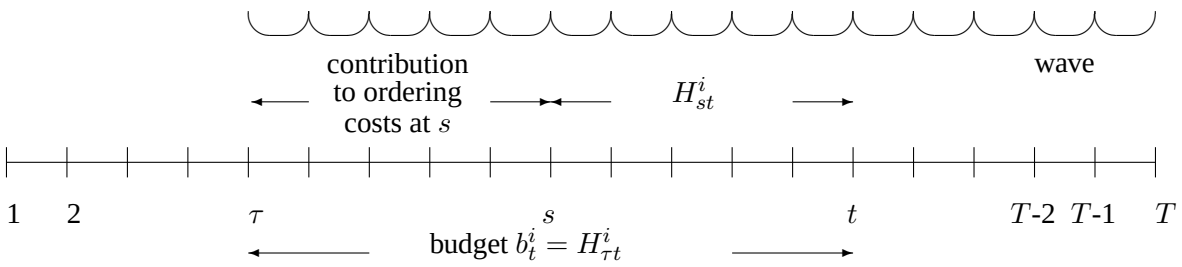

Figure 2.1: The waveform specification of the budget b_t^i and its allocation.

s. Each potential order s collects the budgets of all relevant demand points (i.e., demands at time period s or later), trying to pay for its cost. The cost of an order consists of the joint ordering cost K_0, the item ordering cost K_i for each item i included in the order, and the holding cost for each demand point provided by the order. Note that each demand point is simultaneously making these offers to multiple potential orders, even though it will ultimately be served by exactly one of them; furthermore, more than one of these orders might be opened, and the extent to which these multiple offers are simultaneously accepted is directly linked to the performance guarantee that we will be able to prove.

Once the cost of some joint order s is fully paid for, we are going to *temporarily open* this joint order. This order at time period s will include exactly those items for which the item ordering cost with respect to s has been fully paid. We then freeze the budgets of all demand points that can be served from that order; that is, all unsatisfied demands for those items ordered in time period s for all time periods s or later. We note that the waveform mechanism ensures that the budget of any frozen demand point is enough to pay for the holding cost incurred by satisfying it from the order at s. This phase ends when all budgets are frozen, providing a feasible dual solution and a feasible solution to the JRP. However, this initial solution is too expensive, since the budget of a demand point might be used to pay for the opening of multiple orders.

This leads to the second phase, in which we prune the initial solution to get a cheaper one. For any such order s, we consider the location τ of the wavefront when s was temporarily opened; let $open(s)$ denote this value. We then say that two orders s and r are *dependent* if and only if the intervals $[open(s), s]$ and $[open(r), r]$ intersect. Next we consider the temporarily opened orders from earliest to latest, and *permanently open* an order s if and only if its associated interval does not intersect with the interval associated with any order already permanently opened. Because of the specific waveform mechanism we are using, this ensures that each demand point is committed to pay for the joint ordering cost K_0 of at most one permanently opened order. However, for the JRP, we also need to specify which items are included in each joint order. Thus, additional work is required. We want to make sure that each demand point (i, t) is provided from a joint order that includes item i and such that the holding cost incurred can be paid by the budget b_t^i.

Finally, we introduce a *charging scheme* that specifies how the cost of the solution constructed to the JRP is paid for, using the dual budgets b_t^i. We show that for the JRP, one can pay for the cost of the solution such that no demand point is charged more than twice its budget b_t^i. This implies that the cost of our solution is within twice the optimal cost.

Next we give the LP formulations that underly this algorithm, and then give the details of the first phase of the algorithm in a more precise way. The following is the LP relaxation of a natural integer programming formulation of the JRP:

$$\text{minimize} \quad \sum_{s=1}^{T} y_s^0 K_0 + \sum_{i=1}^{N} \sum_{s=1}^{T} y_s^i K_i + \sum_{i=1}^{N} \sum_{t=1}^{T} \sum_{s=1}^{t} x_{st}^i H_{st}^i \tag{P}$$

$$\text{subject to} \quad \sum_{s=1}^{t} x_{st}^i = 1, \qquad i = 1, \ldots, N,\ t = 1, \ldots, T, \tag{1}$$

$$x_{st}^i \leq y_s^i, \qquad i = 1, \ldots, N,\ t = 1, \ldots, T,\ s = 1, \ldots, t, \tag{2}$$

$$x_{st}^i \leq y_s^0, \qquad i = 1, \ldots, N,\ t = 1, \ldots, T,\ s = 1, \ldots, t, \tag{3}$$

$$x_{st}^i,\ y_s^i,\ y_s^0 \geq 0, \qquad i = 1, \ldots, N,\ s = 1, \ldots, T,\ t = s, \ldots, T. \tag{4}$$

The variable x_{st}^i indicates whether the demand d_{it} was provided from period s. The variable y_s^i indicates whether item i was ordered in period s. The variable y_s^0 indicates if *any* item was ordered in period s. The constraint (1) ensures that each demand point (i, t) is satisfied from some time period $s \leq t$. The constraint (2) ensures that no demand for item i can be provided from period s without placing an order for item i at s. The constraint (3) ensures that no demand can be provided from period s without placing a joint order at s. The integer programming formulation is correct because of the well-known property of the JRP that there exists an optimal solution where each demand point is provided from a single order. The dual (D) of the LP above is:

$$\text{maximize} \quad \sum_{i=1}^{N} \sum_{t=1}^{T} b_t^i \tag{D}$$

$$\text{subject to} \quad b_t^i \leq H_{st}^i + l_{st}^i + z_{st}^i, \qquad i = 1, \ldots, N,\ t = 1, \ldots, T,\ s = 1, \ldots, t. \tag{5}$$

$$\sum_{t=s}^{T} l_{st}^i \leq K_i, \qquad i = 1, \ldots, N,\ s = 1, \ldots, T. \tag{6}$$

$$\sum_{i=1}^{N} \sum_{t=s}^{T} z_{st}^i \leq K_0, \qquad s = 1, \ldots, T, \tag{7}$$

$$l_{st}^i,\ z_{st}^i \geq 0, \qquad i = 1, \ldots, N,\ t = 1, \ldots, T,\ s = 1, \ldots, t. \tag{8}$$

Naturally, (D) provides a lower bound on the cost of any feasible solution to the JRP, since it provides a lower bound on the optimal value of (P), which is itself a lower bound on the optimal cost of the JRP.

As we have already indicated, we think of the dual variable b_t^i as a budget associated with the demand point (i, t). This budget is offered to the various potential orders (i.e., orders $s = 1, \ldots, t$) so as to be served by one of them. Each potential order $s = 1, \ldots, T$ collects the budgets from different relevant demand points so as to fully pay for the cost of its opening. This cost consists of a joint ordering cost K_0, an item ordering cost K_i for each item i included in the order, as well as the holding cost H_{st}^i of each demand point served by the order. When offered to a potential order s, the budget b_t^i is first used to pay for the holding cost incurred by providing d_{it} from period s, namely H_{st}^i. Then the residual budget is used to pay some share of the item ordering cost K_i. The payment of demand point (i, t) towards the item ordering cost at s is captured through the dual variable l_{st}^i. When the item ordering cost is fully paid, demand point (i, t) might pay some share in the joint ordering cost K_0 at s. This is captured through the dual variable z_{st}^i. Thus, with respect to the potential order s, the budget b_t^i is allocated into three different parts, H_{st}^i, l_{st}^i and z_{st}^i.

Next we outline our primal-dual procedure in more detail, and explicitly link the behavior of the algorithm with the LP formulations above. Our procedure is a dual ascent procedure: each dual variable b_t^i is initially equal to 0, and then is only increased until it is frozen at its final value.

As we indicated above, one of the novel ideas in our algorithm is that we do not increase the dual variables uniformly over time, but rather use the waveform mechanism described above. We initialize the wavefront variable τ to T. The algorithm consists of a series of iterations as the value of τ is (continuously) decreased through the interval $[T, 1]$. This parameter controls the values of the budgets b_t^i of each unfrozen demand point (i, t): we have indicated that the budget is always equal to $H_{\tau t}^i$, but this is defined only for integral values of τ. We extend this notion for $\tau \in (s-1, s)$, for some integer s, by simply linearly interpolating the values $H_{s-1,t}^i$ and H_{st}^i.

As the wave moves backward in time, we will *temporarily open joint orders, temporarily add items to joint orders* and *freeze* budgets of demand points; as the budgets are increased we identify the following events:

Event 1 When $\tau = s$ (for $s = T, T-1, \ldots, 1$), we consider all unfrozen demand points (i, t) with $t = s, \ldots, T$ and start increasing the variable l_{st}^i at the same rate as b_t^i. In other words, as long as b_t^i is not frozen (and $\tau \leq t$), we keep $b_t^i = H_{\tau t}^i = H_{st}^i + l_{st}^i$; l_{st}^i is the portion in the budget b_t^i that is used to pay for a share of item ordering cost of item i at time period s. (Note that as the wavefront reaches s and the budget b_t^i increases to H_{st}^i , the constraint (5) becomes tight. As the budget increases further as the wavefront "advances" from s towards $s-1$, in order to continue increasing the budget and remain dual feasible, we must also increase the right-hand side of (5).)

Event 2 Suppose that for some i and s, we have that $\sum_{t \geq s} l_{st}^i = K_i$. (Note that this means that we can no longer increase any variable l_{st}^i without violating the constraint (6).) Then one of the following cases applies:

(a) Suppose that the joint order for time period s is not yet temporarily opened (joint orders will be opened in Event 3, below). Consider all unfrozen demand points (i, t) with $t \geq s$. We freeze the variables l_{st}^i and instead start increasing the variables z_{st}^i (at the same rate as the budget b_t^i). We then have that $b_t^i = H_{\tau t}^i = H_{st}^i + l_{st}^i + z_{st}^i$ (where z_{st}^i accounts for the portion in the budget b_t^i that is used to pay for the joint ordering cost for s).

(b) The joint order at time period s is already temporarily opened. Then we add item i to the order at s and freeze the budgets of all unfrozen demand points (i, t) with $t \geq s$.

Event 3 Suppose that for some period $s > 1$, $\sum_{i=1}^{N} \sum_{t \geq s} z_{st}^i = K_0$. (Note that we can no longer increase any variable z_{st}^i without violating the constraint (7).) Then we declare that the joint order in period s is temporarily opened. In this order at s, we include any item i such that $\sum_{t \geq s} l_{st}^i = K_i$. For each such item i, we freeze the budget of any unfrozen demand point (i, t) with $t \geq s$.

Event 4 Suppose $\tau = 1$. We then open a joint order in period 1. We add to this order all the items $i = 1, .., N$. We then charge the cost of this order to the dual variables of the demand points $(i, 1)$ by setting $b_1^i := l_{11}^i + z_{11}^i$, where $l_{11}^i := K_i$ and $z_{11}^i := K_0/N$ (for $i = 1, .., N$). Next we freeze all the unfrozen budgets and terminate.

We note that the various events described above are likely to occur at non-integer wavefront locations (i.e., for non-integer values of τ). The procedure continues until all budgets are frozen (i.e., until *Event 4* above happens). In case several events happen simultaneously we consider them in an arbitrary order.

Let $(\hat{b}, \hat{l}, \hat{z})$ be the dual solution generated at the end of this phase. It is easily seen that this a feasible dual solution. Moreover, the above procedure also induces a feasible (integer) primal solution. However, this solution is rather expensive, since the budget of a demand point can be multiply used to pay towards several orders. Next, we discuss the second phase of the algorithm, in which we prune the solution to get a

cheaper one in which this overpayment is bounded. We first discuss the simpler special case of the lot-sizing problem (in Section 3) and then discuss the more general model of the JRP (in Section 4). n 4).

3 The single-item lot-sizing problem

In this section, we show that the primal-dual framework produces an optimal solution to the single-item lot-sizing problem. We start with this model, rather than the JRP, since this allows us highlight the main ideas of the algorithm and its analysis. This lot-sizing problem can be viewed as the special case of the JRP in which $N = 1$ and $K_0 = 0$. To simplify our notation, we will only have an ordering cost K and holding costs h_{st}, where we now omit the item index. The primal and dual LPs are also simpler, as follows:

$$\text{minimize} \quad \sum_{s=1}^{T} y_s K \quad + \quad \sum_{t=1}^{T}\sum_{s=1}^{t} x_{st} H_{st} \tag{P1}$$

$$\text{subject to} \quad \sum_{s=1}^{t} x_{st} = 1, \qquad t = 1, \ldots, T, \tag{9}$$

$$x_{st} \le y_s, \qquad t = 1, \ldots, T,\ s = 1, \ldots, t, \tag{10}$$

$$x_{st},\ y_s \ge 0, \qquad s = 1, \ldots, T,\ t = s, \ldots, T. \tag{11}$$

We also get similar dual:

$$\text{maximize} \quad \sum_{t=1}^{T} b_t \tag{D1}$$

$$\text{subject to} \quad b_t \le H_{st} + l_{st}, \qquad t = 1, \ldots, T,\ s = 1, \ldots, t. \tag{12}$$

$$\sum_{t=s}^{T} l_{st} \le K, \qquad s = 1, \ldots, T. \tag{13}$$

$$l_{st} \ge 0, \qquad t = 1, \ldots, T,\ s = 1, \ldots, t. \tag{14}$$

If one considers the primal-dual framework applied to this setting, the budget b_t of any demand point t is allocated (with respect to any order s) to pay for the cost of holding the demand d_t from s to t, and then the leftover amount l_{st} is used to pay a share of the ordering cost at s, K.

We apply the procedure described in Section 2, but now an order s is temporarily opened as soon as its ordering cost K is fully paid, i.e., when $\sum_{t \ge s} l_{st} = K$. Let $(\hat{b}, \hat{l})$ be the dual feasible solution at the end of the first phase. We next describe the pruning phase.

Let $R = \{s_1 = 1 < s_2 < \cdots < s_m\}$ be the set of the time periods of all temporarily opened orders. For each $s \in R$, let $open(s)$ be the location of the wavefront when the order at s was temporarily opened. We say that the interval $[open(s), s]$ is the *shadow interval* of s. Furthermore, r and s in R are said to be *dependent* if and only if their shadow intervals intersect. We consider the periods s_i, $i = 1, \ldots, m$, in increasing order of s_i, and permanently open an order s_j whenever its associated shadow interval does not intersect the shadow interval of any earlier s_i, $i = 1, \ldots, j-1$, that has already been permanently opened. Let $R' \subseteq R$ be the set of time periods of the permanently opened orders. Given the set R', we get a feasible solution to the lot-sizing problem by satisfying each demand from the latest possible order R'. Let $(\hat{x}, \hat{y})$ denote this solution.

3.1 Analysis of the lot-sizing algorithm

We next show that our algorithm finds an optimal solution to the single-item lot-sizing problem. The main idea is to show that we can pay for the cost of $(\hat{x}, \hat{y})$ using the feasible dual budgets $\hat{b}_t$, in such a way that any demand point t is charged exactly its budget $\hat{b}_t$, $t = 1, \ldots, T$.

By the construction of the algorithm we know that for each $s \in R'$ we have $\sum_{t \geq s} l_{st} = K$. We will say that a demand point t *contributes* towards an order $s \in R'$ if $l_{st} > 0$. In addition, each demand point should pay for its holding cost. We use $\hat{H}_t$ to denote the holding cost incurred by demand point t in $(\hat{x}, \hat{y})$, i.e., $\sum_{s=1}^{t} H_{st}\hat{x}_{st}$.

For each demand point $t = 1, \ldots, T$, let $freeze(t)$ be the location of the wavefront τ when its budget was frozen, i.e., $\hat{b}_t = H_{freeze(t),t}$. We call the interval $[freeze(t), t]$ the *active interval* of t. This is the interval along which we increased the budget b_t. Clearly, the demand point t can contribute only towards orders within its active interval only.

Lemma 3.1 *For any demand point $t = 1, .., T$, there exists a single order $s \in R'$ that is within its active interval.*

Proof : We first show that there exists an order $s \in R'$ within the active interval of t. Let $s' \in R$ be the order that caused the budget of t to be frozen. By definition of the specific waveform mechanism we are using, we have that $open(s') = freeze(t)$. If $s' \in R'$, then since s' is in the active interval of t, we are done. Otherwise, there must be some $s \in R'$, with $s < s'$, whose shadow interval intersects the shadow interval of s'. Thus, we have that $freeze(t) = open(s') \leq s < s' \leq t$. But this implies that $s \in [freeze(t), t]$, i.e., s is in the active interval of t.

Next we show that at most one order $s \in R'$ is within $[freeze(t), t]$. Let s now denote the latest order within $[freeze(t), t] \cap R'$. Clearly, $open(s) \leq freeze(t)$, since if the demand t was not frozen until s was temporarily opened, it must have been frozen then. However, since $s \in R'$, it must be the case that $R' \cap [open(s), s) = \emptyset$, since otherwise we would not permanently open s. Since $open(s) \leq freeze(t)$, we see that $R' \cap [freeze(t), s) = \emptyset$, which implies the lemma. ■

As a corollary of this lemma, we get the following theorem:

Theorem 3.2 *The primal-dual algorithm finds an optimal solution to the single-item lot-sizing problem.*

Proof : Lemma 3.1 implies that any demand point t contributes towards exactly one order $s \in R'$. More specifically, the share that demand point t contributes towards this order s is exactly $\hat{l}_{st}$. Moreover, in $(\hat{x}, \hat{y})$, the demand d_t will be satisfied by the order in time period s, and so the holding cost it incurs is equal to H_{st}. Recall that $\hat{b}_t = H_{st} + \hat{l}_{st}$. We get that $\hat{b}_t$ is sufficient to pay for both t's contribution $\hat{l}_{st}$ to the order at s and the holding cost $\hat{H}_t = H_{st}$ incurred by t in $(\hat{x}, \hat{y})$. As a result, we get that the cost of $(\hat{x}, \hat{y})$ is equal to $\sum_t \hat{b}_t$, which implies the theorem. ■

It is important to note that if we generalized the input to allow that the cost of placing an order in period s is a time-dependent parameter K_s, the identical algorithm and analysis yield the the same theorem in this more general setting. etting.

4 The joint replenishment problem

We now describe the second phase of the primal-dual algorithm for the JRP, and give its analysis. This pruning phase is more involved for the JRP than for the lot-sizing problem, since we need to determine not only the time periods at which orders are placed, but also which items are included in each joint order.

Let $R := \{s_1 = 1 < s_2 < \cdots < s_m\}$ be the set of time periods of all temporarily opened joint orders. We extend the terminology introduced in the previous section, to again define $open(s)$ (for orders $s \in R$), $freeze(i,t)$ (for demand points (i,t)) as well as the corresponding shadow and active intervals. In addition, we say that item i is a *contributor* to an order $s \in R$, if it pays a share of the joint ordering cost at s (i.e., $\sum_{t \geq s} \hat{z}^i_{st} > 0$). Let $\mathcal{C}(s)$ be the set of contributor items for some $s \in R$.

We start by applying the same procedure as for the lot-sizing problem to get a subset $R' \subseteq R$ of permanently opened joint orders (i.e., we process the orders in R from earliest to latest, retaining the next only if its shadow interval does not intersect the shadow interval of any order already in R'). Initially, for any joint order $s \in R'$, we include all of its contributor items $i \in \mathcal{C}(s)$. We call these orders *regular orders*.

Again using the properties of the waveform mechanism, it is straightforward to show that each demand point (i,t) has at least one joint order $s \in R'$ within its active interval (by a proof nearly identical to this part of Lemma 3.1). However, there is no guarantee that there is a regular order within the active interval of (i,t) that includes item i. As a result, more work is required.

Focus on one item i, and find the latest demand point (i,t) such that there does not exist a regular order of item i within its active interval. We have already observed that there does exist at least one permanently opened joint order $s \in R'$ within its active interval. Hence, we can add an *extra order* of item i to the *earliest* joint order $s \in R' \cap [freeze(i,t), t]$. We shall say that (i,t) is the *initiator* of the extra order of item i in period s. This process is repeated on the remaining time horizon $[1, s)$, and continues until each demand point (i,t) can be served, by either a regular or extra order, within its active interval. The same procedure is repeated for each item i.

After all the orders are specified, each demand point (i,t) is then satisfied from the latest possible period $s \in R'$ containing item i. Let $(\hat{x}, \hat{y})$ denote the solution found for the JRP.

4.1 Analysis of the JRP algorithm

We will show that the cost of $(\hat{x}, \hat{y})$ can be paid using the dual feasible budgets $(\hat{b}, \hat{l}, \hat{z})$ such that each demand point (i,t) is charged at most $2\hat{b}^i_t$. For this, we need to introduce a somewhat more involved charging scheme.

For the regular orders and the joint ordering cost, we use the contributor items to pay for both their joint and item ordering cost. The ordering cost of any regular order $s \in R'$ is $K_0 + \sum_{i \in \mathcal{C}(s)} K_i = \sum_{i \in \mathcal{C}(s)} \sum_{t \geq s} (\hat{l}^i_{st} + \hat{z}^i_{st})$. This follows from the construction of the algorithm.

Now consider an extra order of item i in period $s \in R'$, and let (i,t) be the initiator of this extra order. Let s' be the freezing order of (i,t), and so $freeze(i,t) \leq open(s')$. By definition, s is the earliest order within $R' \cap [freeze(i,t), t]$. Also observe that $R' \cap [open(s'), s'] \neq \emptyset$, since either $s' \in R'$, or it was eliminated by some earlier order $s'' \in R'$, such that $open(s') \leq s'' < s'$. Consequently, $s \leq s'$. Since s' is the freezing order of demand point (i,t), it follows that $\sum_{t \geq s'} \hat{l}^i_{s't} = K_i$; we can use this to pay for the cost of the extra order of item i at s. To indicate this connection, we will denote s' by $\mathsf{N}_i(s)$. Here we use the fact that the item ordering cost K_i is the same for each time period.

Consider any demand point (i,t); we will say that (i,t) *contributes towards some regular order* $s \in R'$ if $i \in \mathcal{C}(s)$ and $\hat{z}^i_{st} + \hat{l}^i_{st} > 0$. In addition, we will say that (i,t) *contributes towards some extra order* of item i in period $s \in R'$ if $\hat{l}^i_{\mathsf{N}_i(s),t} > 0$. Thus, we charge demand point (i,t) with what it contributes towards different orders in R' as well as the holding cost it incurs in $(\hat{x}, \hat{y})$. Denote this holding cost by $\hat{H}^i_t$.

We now show that, using the above charging scheme, one can pay for the cost of $(\hat{x}, \hat{y})$ such that no demand point is charged more than $2\hat{b}^i_t$. We first state and prove the following lemma, which is central to our result:

Lemma 4.1 *Consider any demand point (i,t) and let $r_1 \in R'$ be the latest order in R', regular or extra, towards which (i,t) contributes. Then, either $r_1 \notin [freeze(i,t), t]$ or it is the earliest order in $R' \cap$*

$[freeze(i,t),t]$.

Proof : Assume $r_1 \in [freeze(i,t),t]$ and consider the following two possible cases:

Case 1. The order in period r_1 is a regular order of item i. We will argue that $open(r_1) \leq freeze(i,t)$. We know that $i \in \mathcal{C}(r_1)$, and so $\sum_{u \geq r_1} \hat{z}^i_{r_1,u} > 0$. By the construction of the waveform, we know that the demand points of an item can start paying a share of the joint ordering cost only after the item ordering cost is fully paid. Thus, when the order at r_1 was temporarily opened, we immediately added item i to that order. Consider the wavefront position τ when the order r_1 is opened (i.e., the wavefront is located in $open(r_1)$); if the demand point (i,t) is not frozen prior to this point in the execution of the algorithm (i.e., when τ is larger), it must become frozen now. In other words, $open(r_1) \leq freeze(i,t)$. By the choice of r_1, we know that its shadow interval $[open(r_1), r_1]$ does not contain another order $r \in R'$. Since $[freeze(i,t), r_1] \subseteq [open(r_1), r_1]$, this implies that r_1 is the earliest order in $R' \cap [freeze(i,t),t]$.

Case 2. The order in r_1 is an extra order of item i. This order has some initiator (i,t^*) with a freezing order $\mathsf{N}_i(r_1)$ such that $r_1 \leq \mathsf{N}_i(r_1) \leq t$. In particular, by the waveform properties we know that $freeze(i,t^*) \leq freeze(i,t)$, since (i,t) was frozen no later than (i,t^*) was (as $\mathsf{N}_i(r_1) \leq t$). However, from the way we add extra orders, we know that the order at r_1 is the earliest in R' within the active interval of the initiator (i,t^*). In other words, $R' \cap [freeze(i,t^*), r_1) = \emptyset$. Given that we already concluded that $freeze(i,t^*) \leq freeze(i,t)$, the lemma follows. ■

The above lemma has several immediate corollaries:

Corollary 4.2 *Any demand point (i,t) can contribute towards at most two orders in R'.*

Proof : Suppose that (i,t) contributes towards more than one order in R', and let $r_1 > r_2$ be the two latest such orders.

Suppose that $r_1 < freeze(i,t)$; in that case, r_1 and r_2 must both be extra orders of item i (since they do not lie in the active interval of (i,t)), and we will argue that (i,t) cannot contribute to both. If (i,t) contributes to r_2, then we must have that $\hat{l}^i_{\mathsf{N}_i(r_2),t} > 0$, and so $r_1 < freeze(i,t) \leq \mathsf{N}_i(r_2)$. But the initiator of r_2 is earlier than r_1 and hence earlier than $\mathsf{N}_i(r_2)$, which is its freezing order. Clearly, it is impossible for this to be true. Hence, (i,t) cannot contribute to more than one extra order that precedes its active interval.

Hence, $r_1 \in [freeze(i,t),t]$. By Lemma 4.1, it follows that r_1 is the earliest permanent order in $[freeze(i,t),t] \cap R'$. Hence, no other order that (i,t) contributes to is within its active interval. Any order to which (i,t) contributes that precedes its active interval is an extra order. But we have already seen that there is at most one such order (namely r_2), which completes the proof. ■

Corollary 4.3 *Consider a demand point (i,t) and let r_1 be the latest order towards which (i,t) contributes some positive share. Then the holding cost that (i,t) incurs in $(\hat{x},\hat{y})$ is at most $H^i_{r_1,t}$ (i.e., $\hat{H}^i_t \leq H^i_{r_1,t}$).*

Proof : Since the algorithm ensures that each demand point (i,t) is satisfied from some order $r \in R'$ within its active interval, the claim follows immediately from Lemma 4.1, since r_1 is either the earliest in $[freeze(i,t),t] \cap R'$ or $r_1 < freeze(i,t)$. ■

We now ready to prove the main theorem:

Theorem 4.4 *The primal-dual framework yields a 2-approximation algorithm for the JRP.*

Proof : Consider any demand point (i,t) and let $r_1 \in R'$ again be the latest order in $(\hat{x},\hat{y})$ towards which (i,t) contributes a positive share. If the order in period r_1 is a regular order of item i, then (i,t) contributes $\hat{l}^i_{r_1,t} + \hat{z}^i_{r_1,t} > 0$. If the order at r_1 is an extra order of item i, then (i,t) contributes $\hat{l}^i_{\mathsf{N}_i(r_1),t} > 0$, where $freeze(i,t) \leq \mathsf{N}_i(r_1) \leq t$. In either case, this is clearly bounded by $\hat{b}^i_t$.

Now assume that (i,t) also contributes towards a second (earlier) order r_2. By Lemma 4.1, r_2 must be an extra order of item i such that $r_2 \notin [freeze(i,t),t]$. Hence, (i,t) contributes $\hat{l}^i_{\mathsf{N}_i(r_2),t} > 0$ towards r_2, where $freeze(i,t) \leq \mathsf{N}_i(r_2) < r_1$. The latter inequality follows from the fact that since item i is included in r_1, the initiator of r_2 is earlier than r_1, and hence so is its freezing order, $\mathsf{N}_i(r_2)$. We have $\hat{b}^i_t = H^i_{\mathsf{N}_i(r_2),t} + \hat{l}^i_{\mathsf{N}_i(r_2),t} + \hat{z}^i_{\mathsf{N}_i(r_2),t} \geq H^i_{\mathsf{N}_i(r_2),t} + \hat{l}^i_{\mathsf{N}_i(r_2),t} \geq H^i_{r_1,t} + \hat{l}^i_{\mathsf{N}_i(r_2),t}$; the first inequality follows from $\hat{z}^i_{\mathsf{N}_i(r_2),t} \geq 0$, and the second inequality follows from the monotonicity of the holding costs and $\mathsf{N}_i(r_2) < r_1$. From Corollary 4.3, we get that $\hat{b}^i_t \geq \hat{H}^i_t + \hat{l}^i_{\mathsf{N}_i(r_2),t}$. Corollary 4.2 also implies that (i,t) does not contribute towards any other order $r \in R'$ other than r_1 and r_2. As a result, we get that the sum of the holding cost incurred by (i,t) and its contributions towards ordering costs is bounded by $2\hat{b}^i_t$. This proves the theorem. ■

We note that the above analysis remains valid if we allow the joint ordering cost (K_0) to be time-dependent. We can also allow time-dependent item ordering costs provided that they are non-decreasing over time. If we allow arbitrary cost parameters, then there exists a simple reduction from the set cover problem, and hence, one can not hope for a constant performance guarantee.

4.2 The JRP With Back Orders

In this section, we consider the extension of the JRP in which back orders are allowed. More specifically, demands in period t *can* be satisfied from orders later in time (i.e., from orders in periods $s > t$). Given a demand d_{it}, we let B^i_{st} be the back order cost of providing this demand from an order in period s, where $s > t$. As before, we will assume that B^i_{st} is non-negative, linear in d_{it} and non-decreasing in $s \geq t$ for any fixed (i,t). We will show that our general assumptions on the holding cost imply that this more general case with back orders can be reduced to the previous variant without back orders.

Consider now any two consecutive orders of item i, say, in periods $s_1 < s_2$. It is easy to compute the optimal policy to minimize the overall holding and back order cost of item i over the interval $[s_1, s_2)$. The monotonicity assumptions imply that each demand point (i,t') with $t' \in [s_1, s_2)$ will be served either from s_1 or from s_2 as a back order. Let G^i_{st} denote the optimal holding and back order cost of item i over $[s,t)$, given that we have two consecutive orders in $s < t$. Observe that G can be computed efficiently for each item i and pair $s < t$. More specifically, for each $t' \in [s,t)$ we only need to consider $\min\{H^i_{s,t'}, B^i_{t,t'}\}$.

We now let $\bar{H}^i_{st} := G^i_{s,t+1} - G^i_{st}$ for each $s < t$, and let $\bar{H}^i_{ss} := H^i_{ss} = B^i_{ss}$. The parameter $\bar{H}^i_{st}$ accounts for the difference in the overall holding and back order cost if instead of ordering item i in s and then in t, we order i in s and next in $t+1$. Because of the monotonicity assumptions, we know that the $\bar{H}$ parameters are non-negative. Using this, we consider the LP in Section 2 having $\bar{H}$ as the objective function coefficients of the x^i_{st} variables (instead of the H parameters). The variable x^i_{st} would now indicate that s is the order of item i closest to t in the interval $[1,t]$. We associate the cost $\bar{H}^i_{st}$ with it, since it is clear that if $x^i_{st} = 1$, then we will have no orders of item i over $(s,t]$.

Next we show that for any fixed (i,t), $\bar{H}^i_{st}$ is non-increasing in s, i.e., it has the same monotonicity property assumed throughout this paper. Hence, we establish the correctness of the new formulation (with the $\bar{H}$ parameters) for the JRP with back orders. Since this monotonicity property was the only assumption needed for the execution of the algorithm and its analysis, we obtain a 2-approximation for this more general model as well. Naturally, this extends the optimality result for the lot-sizing case as described in Section 3. We believe that this is the first primal-dual algorithm for this variant of the lot-sizing problem.

Lemma 4.5 *Consider some demand point (i,t) and some $1 < s < t$. Then $\bar{H}^i_{st} \leq \bar{H}^i_{s-1,t}$.*

Proof : For each demand point (i,t') and some $s_1 \leq t' < s_2$, we let $\Delta^{it'}_{s_1,s_2}$ be the difference between the cheaper of the holding or back order costs for (i,t') for the interval $[s_1,s_2)$ (i.e., $\min\{H^i_{s_1,t'}, B^i_{s_2,t'}\}$), and the cheaper of the holding or back order costs for the interval $[s_1,s_2+1)$. In other words, $\bar{H}^i_{s_1,s_2} = \sum_{t' \in [s_1,s_2)} \Delta^{it'}_{s_1,s_2}$. Focus now on some demand point (i,t') with $t' \in [s,t]$. By the monotonicity assumption we know that $\Delta^{it'}_{st} \geq 0$. It is sufficient to show that $\Delta^{it'}_{st} \leq \Delta^{it'}_{s-1,t}$. Consider the optimal solutions for (i,t') for the intervals $[s,t)$ and $[s,t+1)$ respectively. There are only three possible cases:

Case 1. Demand point (i,t') is served from s in the optimal solutions for both intervals. In this case, we have $\Delta^{it'}_{st} = 0$, and the claim follows immediately.

Case 2. Demand point (i,t') is served as a back order in the optimal solutions for both intervals. Observe that the monotonicity assumption implies that (i,t') is served as a back order also in the optimal solutions for the intervals $[s-1,t)$ and $[s-1,t+1)$, respectively. Hence, $\Delta^{it'}_{st} = \Delta^{it'}_{s-1,t} = B^i_{t+1,t'} - B^i_{tt'}$, and again the claim follows.

Case 3. Demand point (i,t') is served as a back order in the optimal solution for $[s,t)$ and from s in the optimal solution for $[s,t+1)$. Using again the monotonicity assumptions, we conclude that (i,t') is served as a back order in the optimal solution for $[s-1,t)$. In addition, we know that $H^i_{st'} < B^i_{t+1,t'}$, since otherwise (i,t') would not switch to s in the optimal solution for $[s,t+1)$. We get that $\Delta^{it'}_{s-1,t}$ is equal to $B^i_{t+1,t'} - B^i_{tt'}$ or to $H^i_{s-1,t'} - B^i_{tt'}$. In either cases $\Delta^{it'}_{s-1,t} \geq \Delta^{it'}_{s-1,t} = H^i_{st'} - B^i_{tt'}$.

This completes the proof. ■

Corollary 4.6 *The primal-dual algorithm provides a 2-approximation algorithm for the JRP with back orders.*

Corollary 4.7 *The primal-dual algorithm solves optimally the single-item lot-sizing problem with back orders.*

4.3 The JRP With Back Orders

In this section, we consider the extension of the JRP in which back orders are allowed. More specifically, demands in period t *can* be satisfied from orders later in time (i.e., from orders in periods $s > t$). Given a demand d_{it}, we let B^i_{st} be the back order cost of providing this demand from an order in period s, where $s > t$. As before, we will assume that B^i_{st} is non-negative, linear in d_{it} and non-decreasing in $s \geq t$ for any fixed (i,t). We will show that our general assumptions on the holding cost imply that this more general case with back orders can be reduced to the previous variant without back orders.

Consider now any two consecutive orders of item i, say, in periods $s_1 < s_2$. It is easy to compute the optimal policy to minimize the overall holding and back order cost of item i over the interval $[s_1,s_2)$. The monotonicity assumptions imply that each demand point (i,t') with $t' \in [s_1,s_2)$ will be served either from s_1 or from s_2 as a back order. Let G^i_{st} denote the optimal holding and back order cost of item i over $[s,t)$, given that we have two consecutive orders in $s < t$. Observe that G can be computed efficiently for each item i and pair $s < t$. More specifically, for each $t' \in [s,t)$ we only need to consider $\min\{H^i_{s,t'}, B^i_{t,t'}\}$.

We now let $\bar{H}^i_{st} := G^i_{s,t+1} - G^i_{st}$ for each $s < t$, and let $\bar{H}^i_{ss} := H^i_{ss} = B^i_{ss}$. The parameter $\bar{H}^i_{st}$ accounts for the difference in the overall holding and back order cost if instead of ordering item i in s and

then in t, we order i in s and next in $t+1$. Because of the monotonicity assumptions, we know that the $\bar{H}$ parameters are non-negative. Using this, we consider the LP in Section 2 having $\bar{H}$ as the objective function coefficients of the x^i_{st} variables (instead of the H parameters). The variable x^i_{st} would now indicate that s is the order of item i closest to t in the interval $[1,t]$. We associate the cost $\bar{H}^i_{st}$ with it, since it is clear that if $x^i_{st}=1$, then we will have no orders of item i over $(s,t]$.

Next we show that for any fixed (i,t), $\bar{H}^i_{st}$ is non-increasing in s, i.e., it has the same monotonicity property assumed throughout this paper. Hence, we establish the correctness of the new formulation (with the $\bar{H}$ parameters) for the JRP with back orders. Since this monotonicity property was the only assumption needed for the execution of the algorithm and its analysis, we obtain a 2-approximation for this more general model as well. Naturally, this extends the optimality result for the lot-sizing case as described in Section 3. We believe that this is the first primal-dual algorithm for this variant of the lot-sizing problem.

Lemma 4.8 *Consider some demand point (i,t) and some $1<s<t$. Then $\bar{H}^i_{st} \leq \bar{H}^i_{s-1,t}$.*

Proof : For each demand point (i,t') and some $s_1 \leq t' < s_2$, we let $\Delta^{it'}_{s_1,s_2}$ be the difference between the cheaper of the holding or back order costs for (i,t') for the interval $[s_1,s_2)$ (i.e., $\min\{H^i_{s_1,t'}, B^i_{s_2,t'}\}$), and the cheaper of the holding or back order costs for the interval $[s_1,s_2+1)$. In other words, $\bar{H}^i_{s_1,s_2} = \sum_{t'\in[s_1,s_2)} \Delta^{it'}_{s_1,s_2}$. Focus now on some demand point (i,t') with $t'\in[s,t]$. By the monotonicity assumption we know that $\Delta^{it'}_{st} \geq 0$. It is sufficient to show that $\Delta^{it'}_{st} \leq \Delta^{it'}_{s-1,t}$. Consider the optimal solutions for (i,t') for the intervals $[s,t)$ and $[s,t+1)$ respectively. There are only three possible cases:

Case 1. Demand point (i,t') is served from s in the optimal solutions for both intervals. In this case, we have $\Delta^{it'}_{st}=0$, and the claim follows immediately.

Case 2. Demand point (i,t') is served as a back order in the optimal solutions for both intervals. Observe that the monotonicity assumption implies that (i,t') is served as a back order also in the optimal solutions for the intervals $[s-1,t)$ and $[s-1,t+1)$, respectively. Hence, $\Delta^{it'}_{st} = \Delta^{it'}_{s-1,t} = B^i_{t+1,t'} - B^i_{tt'}$, and again the claim follows.

Case 3. Demand point (i,t') is served as a back order in the optimal solution for $[s,t)$ and from s in the optimal solution for $[s,t+1)$. Using again the monotonicity assumptions, we conclude that (i,t') is served as a back order in the optimal solution for $[s-1,t)$. In addition, we know that $H^i_{st'} < B^i_{t+1,t'}$, since otherwise (i,t') would not switch to s in the optimal solution for $[s,t+1)$. We get that $\Delta^{it'}_{s-1,t}$ is equal to $B^i_{t+1,t'} - B^i_{tt'}$ or to $H^i_{s-1,t'} - B^i_{tt'}$. In either cases $\Delta^{it'}_{s-1,t} \geq \Delta^{it'}_{s-1,t} = H^i_{st'} - B^i_{tt'}$.

This completes the proof. ■

Corollary 4.9 *The primal-dual algorithm provides a 2-approximation algorithm for the JRP with back orders.*

Corollary 4.10 *The primal-dual algorithm solves optimally the single-item lot-sizing problem with back orders.*

5 Assembly Problem

In this section, we present the required modifications in order to apply the primal-dual method to the assembly problem. Recall that the assembly problem can be presented as a rooted directed in-tree, where each node in the tree corresponds to an item. We also assume that the items are indexed so that $i > j$ for each edge (i, j) in the tree. Item 1, the root of the tree, is facing external demand over T time periods ($d_1, .., d_T$). The idea is that any unit of item i is assembled from one unit of each of its direct predecessor items in the tree. We let $\mathcal{P}(i)$ and $\mathcal{S}(i)$, respectively, be the set of *all* predecessors and successors of item i within the in-tree (both including item i). Furthermore, let $\mathcal{P}'(i)$ denote the direct predecessors of item i, and we let $\sigma(i)$ be its direct successor. Finally, for each item i and each item $k \in \mathcal{P}(i)$, we let $path_{ki}$ be the path from k to i ($k > i$) in the tree defined above.

5.1 A Linear Program

We start by explaining how one can formulate the assembly problem as an integer program with a structure similar to that exploited for the JRP. For this, we need to introduce some well-known results from inventory theory. In multi-stage models such as the assembly problem, it is often more convenient to consider the *echelon inventory level*, as opposed to the *conventional inventory level* discussed previously. The echelon inventory level of item i is defined to be the overall number of units of that item in the system, which includes units that are assembled into other items. Thus, the echelon inventory level of item i is equal to the sum of the conventional inventory levels of all items in $\mathcal{S}(i)$. Given the conventional holding cost parameters h^i_{st}, one can compute the echelon holding cost parameters as $\bar{h}^i_{st} := h^i_{st} - \sum_{k\in\mathcal{P}'(i)} h^k_{st}$, i.e., as the marginal additional conventional holding cost due to assembling item i. We again assume that $\bar{h}^i_{st}$ is non-negative and monotone in s for any fixed (i, t).

One well-known result on the assembly problem is the optimality of what is called the class of *nested policies* (see [8]). In a nested policy, whenever we place an order of item i, we simultaneously place an order for its direct successor item in the tree, $\sigma(i)$. In other words, we can assume that we place an order for item i at time period s only if we also place an order for every item $j \in \mathcal{S}(i)$ at the same time period. Finally, the assembly problem is also known to have an optimal policy such that each demand is provided from a single order.

By relying on the properties stated above, it is straightforward to adapt the linear programming relaxation given in Section 2 to the assembly problem:

$$\text{minimize} \qquad \sum_{i=1}^{N}\sum_{s=1}^{T} y^i_s K_i + \sum_{i=1}^{N}\sum_{t=1}^{T}\sum_{s=1}^{t} x^i_{st}\bar{H}^i_{st} \tag{P2}$$

$$\text{subject to} \qquad \sum_{s=1}^{t} x^i_{st} = 1, \qquad i = 1,\dots,N,\ t = 1,\dots,T, \tag{15}$$

$$x^i_{st} \le y^j_s, \qquad i = 1,\dots,N,\ t = 1,\dots,T,\ s = 1,\dots,t,\ j \in \mathcal{S}(i) \tag{16}$$

$$x^i_{st},\ y^i_s \ge 0, \qquad i = 1,\dots,N,\ s = 1,\dots,T,\ t = s,\dots,T. \tag{17}$$

There no longer is a joint ordering cost, so the variables y^0_s are eliminated, along with their terms in the objective function, as well as the constraints (3). The objective function coefficient of the assignment variables x^i_{st} is the corresponding echelon holding cost $\bar{H}^i_{st}$. Finally, one has the constraint that $x^i_{st} \le y^j_s$ for each $j \in \mathcal{S}(i)$ (and for each period $s \le t$). This implies the nestedness property. Note that in the above LP there are many redundant constraints. However, since we are not going to solve the LP, it does not have any

impact. On the other hand we get a "nicer" dual problem:

$$\text{maximize} \quad \sum_{i=1}^{N}\sum_{t=1}^{T} b_t^i \tag{D2}$$

$$\text{subject to} \quad b_t^i \le \bar{H}_{st}^i + \sum_{j\in\mathcal{S}(i)} z_{st}^{ij}, \qquad i=1,\dots,N,\ t=1,\dots,T,\ s=1,\dots,t. \tag{18}$$

$$\sum_{k\in\mathcal{P}(i)}\sum_{t\ge s} z_{st}^{ki} \le K_i, \qquad i=1,\dots,N,\ s=1,\dots,T. \tag{19}$$

$$z_{st}^{ij} \ge 0, \qquad i=1,\dots,N,\ t=1,\dots,T,\ s=1,\dots,t,\ j\in\mathcal{S}(i) \tag{20}$$

5.2 Primal-Dual Procedure

We use a similar procedure to construct the dual solution and the initial feasible (integer) primal solution. In particular, we use again the waveform mechanism and keep any unfrozen budgets for periods $t \ge \tau$ so that they satisfy $b_t^i = \bar{H}_{\tau t}^i$. We note that here a demand point (i,t) corresponds to providing d_t units of item i to node 1 (i.e., item 1), so as to satisfy the external demand d_t. Given a potential order s, the budget b_t^i will be allocated to $\bar{H}_{st}^i + \sum_{j\in\mathcal{S}(i)} z_{st}^{ij}$. More specifically, it will be used to pay for the echelon holding cost incurred by holding d_t units of item i in the system from period s to t, as well as to possibly contribute a share of the item ordering cost at s of items $j \in \mathcal{S}(i)$ (through the variables z_{st}^{ij}). Of course, we must also maintain, for each item i and each ordering period s, that the total of the shares contributed, $\sum_{t\ge s}\sum_{k\in\mathcal{P}(i)} z_{st}^{ki} \le K_i$. We will temporarily open an order in period s only when the ordering cost of item 1 at s is fully paid. We will add item i to this order only if each item on the path from i to item 1 (i.e., each $j \in path_{i1}$) has already fully paid for its item ordering cost with respect to s.

We now describe the first phase of the algorithm in detail, focusing on the different events that may occur:

Event 1 When $\tau = s$ (for $s = T, T-1, \dots, 2$), we consider all unfrozen demand points (i,t) with $t = s,\dots,T$ and start increasing the variable z_{st}^{ii} at the same rate as b_t^i (keeping $b_t^i := \bar{H}_{\tau t}^i = \bar{H}_{st}^i + z_{st}^{ii}$).

Event 2 Suppose that for some item $i > 1$ and some period $s > 1$, we have that $\sum_{k\in\mathcal{P}(i)}\sum_{t\ge s} z_{st}^{ki} = K_i$. (Note that this means that we can no longer to continue to increase any of the variables z_{st}^{ki} without violating the constraint (19) of item i.) Then one of the following cases applies:

(a) Suppose that the order in time period s is already temporarily opened (see Event 3 below) and includes *all items* $j \in \mathcal{S}(i)\setminus\{i\}$. Then we add to this order each item $k \in \mathcal{P}(i)$ with a positive contribution towards the item ordering cost of item i at s, i.e., the set of items $\{k \in \mathcal{P}(i) : \sum_{t\ge s} z_{st}^{ki} > 0\}$. Note that all of these items have the property that each $j \in path_{ki}$ has already fully paid for its item ordering cost K_j with respect to s. For each such item k, we then freeze the budget of any unfrozen demand point (k,t) with $t \ge s$.

(b) Otherwise, consider the item $j \in \mathcal{S}(i)$ with highest index, such that its item ordering cost is not yet fully paid. Let j' be that item. Each item that has a positive contribution towards the item ordering cost of item i at s will now start to contribute towards the item ordering cost of that item j' at s. More precisely, let $j' := \max\{j \in \mathcal{S}(i) : \sum_{k\in\mathcal{P}(j)}\sum_{t\ge s} z_{st}^{kj} < K_j\}$; clearly, $1 \le j' < i$. Then, for each item $k \in \mathcal{P}(i)$ with $\sum_{t\ge s} z_{st}^{ki} > 0$, consider each unfrozen demand point (k,t) with $t \ge s$: freeze

the variable z_{st}^{ki} and instead start increasing the variable $z_{st}^{kj'}$ (at the same rate as the budget b_t^i). The variable $z_{st}^{kj'}$ accounts for the portion in the budget b_t^k that is used to pay a share towards the ordering cost $K_{j'}$ of item j' with respect to s.

Event 3 Suppose that for some period $s > 1$, $\sum_{k=1}^{N} \sum_{t \geq s} z_{st}^{k1} = K_1$. (Note that we can no longer increase any variable z_{st}^{k1} without violating the constraint (19) with respect to item 1, the root of the tree.) Then we declare that the order in period s is temporarily opened. We add to this order at s any item i such that each item $j \in \mathcal{S}(i)$ has fully paid for their item ordering cost K_j at s, i.e., that for each item $j \in \mathcal{S}(i)$, we have $\sum_{k \in \mathcal{P}(j)} \sum_{t \geq s} z_{st}^{kj} = K_j$. For each such item i, we freeze the budget of each unfrozen demand point (i,t) with $t \geq s$.

Event 4 Suppose $\tau = 1$. We then open the order in period 1. We add to this order all of the items $i = 1, .., N$. We then charge the cost of this order to the dual variables of the demand points $(i, 1)$ by setting $b_1^i := z_{11}^{ii} := K_i$ (for $i = 1, .., N$). Next we freeze all of the unfrozen budgets and terminate.

The solution $(\hat{b}, \hat{z})$ at the end of this phase is clearly dual feasible with respect to (D2). However, the initial (primal) solution for the assembly problem is again potentially too expensive, so we need again to prune it.

5.3 The Pruning Phase

We perform the pruning phase in an iterative way, starting at item 1 and then considering its predecessors. We treat item i only when all of the orders of its successor items are already permanently determined. Let $R := \{s_1 = 1 < s_2 .. < s_m\}$ be the set of the time periods of all temporarily opened orders at the end of the first phase. For the presentation of the pruning phase and the analysis of the algorithm, we introduce an extended notion of the *contributor items*. Consider an order of item i at time period s; we will say that item $k \in \mathcal{P}(i)$ is a contributor item to this order if $\sum_{t \geq s} \hat{z}_{st}^{ki} > 0$. We will denote the set of contributor items by $\mathcal{C}(i, s)$. We again use $open(s)$ and the corresponding shadow interval (for any $s \in R$) and $freeze(i, t)$ and the corresponding active interval (for any (i, t)).

We start with item 1, and perform the same greedy procedure as before to compute a subset $R' \subseteq R$ of permanently opened orders; i.e., we process the orders in R from earliest to latest, retaining the next only if its shadow interval does not intersect the shadow interval of any order already in R'. For each order $s \in R'$, we initially add all of the contributor items $i \in \mathcal{C}(1, s)$, and call these *regular orders*.

Next we consider the rest of the items $i = 2, .., N$ in a way such that each item i is considered only after $\sigma(i)$ was considered. Focus now on some item $i > 1$, we perform a similar procedure to the one described for the JRP in Section 4. We start at T and look for the first demand point, say (i, t), such that there does not exist an order (either regular or extra) of item i within its active interval, $[freeze(i, t), t]$. Let $s' \in R$ be its freezing order. We now consider the earliest order in $R' \cap [freeze(i, t), t]$ *with item* $\sigma(i)$, say s, and add to this order all of the contributor items of the order of i at s', $k \in \mathcal{C}(i, s')$. Observe that for each $k \in \mathcal{C}(i, s')$, it is also the case that each item k' on the path from k to i (i.e., $k' \in path_{ki}$) is also a contributor item (i.e., $k' \in \mathcal{C}(i, s')$). We call these orders *extra orders*. We say that (i, t) and i are the *initiator* and the *initiator item*, respectively, of these extra orders in s. As before, denote $s' := \mathsf{N}_i(s)$. We then continue iteratively on $[1, s)$, until each demand point (i, t) has a permanently open order with item i within its active interval.

We now argue why the above procedure is well defined, and moreover that $s \leq s'$. Observe that for item i such that $\sigma(i) = 1$, the arguments are identical to the ones in the JRP case (see Section 4). So, for each i, we can assume by induction that the procedure is well defined for $\sigma(i)$. Consider now the demand point

$(\sigma(i), s')$; we claim that $freeze(i,t) \leq freeze(\sigma(i), s')$. Recall that (i,t) was frozen just when item i was added to the order at s'; hence item $\sigma(i)$ must have been added to s' either with item i, or perhaps earlier. In particular, $(\sigma(i), s')$ was frozen either with (i,t) or even earlier, i.e., $freeze(i,t) \leq freeze(\sigma(i), s')$. By induction, we know that when (i,t) is considered, we have already ensured that there exists a permanently open order in $R' \cap [freeze(\sigma(i), s'), s']$ with item $\sigma(i)$. Since $[freeze(\sigma(i), s'), s'] \subseteq [freeze(i,t), t]$, we conclude that the procedure described above is indeed well-defined and $s \leq s'$.

It is now clear that at the end of the pruning phase, we have a feasible nested solution to the assembly problem. Let $(\hat{x}, \hat{y})$ be this solution. Next we will show that the cost of the solution is no more than twice the optimal cost.

5.4 Analysis of The Assembly Problem

We start by describing a charging scheme of how the cost of $(\hat{x}, \hat{y})$ can be paid using the feasible dual budgets $(\hat{b}, \hat{z})$. For any order $s \in R'$, let $I(s)$ be the set of the initiator items of the extra orders included in s in $(\hat{x}, \hat{y})$. We pay for the ordering cost of the regular orders at s, i.e., of items $i \in \mathcal{C}(1,s)$, using $\sum_{i \in \mathcal{C}(1,s)} \sum_{j \in \mathcal{S}(i)} \sum_{t \geq s} \hat{z}_{st}^{ij} = \sum_{i \in \mathcal{C}(1,s)} K_i$. The equality is correct based on the observation that if for some $k \in \mathcal{P}(i)$ and $j \in \mathcal{S}(i)$ we have $k \in \mathcal{C}(i,s)$ and $i \in \mathcal{C}(j,s)$, then we also have $k \in \mathcal{C}(j,s)$.

As for the extra orders in s, we can partition them according to their initiator item in $I(s)$. Thus, we have $\sum_{i \in I(s)} \sum_{k \in \mathcal{C}(i, \mathsf{N}_i(s))} \sum_{l \in path_{ki}} \sum_{t \geq \mathsf{N}_i(s)} \hat{z}_{\mathsf{N}_i(s),t}^{kl} = \sum_{i \in I(s)} \sum_{k \in \mathcal{C}(i, \mathsf{N}_i(s))} K_k$. This is correct based on the construction of the algorithm and the same argument used above for the regular orders.

For each demand point (i,t) we say that it contributes towards a regular order in period $s \in R'$ if $i \in \mathcal{C}(1,s)$ and $\sum_{j \in \mathcal{S}(i)} \hat{z}_{st}^{ij} > 0$. We say that (i,t) contributes towards extra orders at some $s \in R'$, if $i \in \mathcal{C}(j, \mathsf{N}_j(s))$ for some $1 < j \in I(s)$ and $\sum_{k \in path_{ij}} \hat{z}_{\mathsf{N}_j(s),t}^{ik} > 0$. In addition, each demand point is charged with the echelon holding cost that it incurs in $(\hat{x}, \hat{y})$; denote this cost by $\hat{H}_t^i$. An important observation is that any demand point (i,t) can only contribute to the opening of orders $s \in R'$ that include item i (either as regular or extra orders).

We are now ready to show that, as in the case of the JRP, one can use the above charging scheme to pay for the cost of $(\hat{x}, \hat{y})$ in a way such that no demand point (i,t) is charged more than twice its budget $\hat{b}_t^i$.

The following are the analogous results to Lemma 4.1 and Corollaries 4.2 and 4.3:

Lemma 5.1 *Consider any demand point (i,t) and let $r_1 \in R'$ be the latest order in R', regular or extra, towards which (i,t) contributes. Then, either $r_1 \notin [freeze(i,t), t]$ or it is the earliest order in $R' \cap [freeze(i,t), t]$ with item i.*

Proof : Assume $r_1 \in [freeze(i,t), t]$ and consider again the following two possible cases:
Case 1. The order of item i in period r_1 is a regular order. In particular, we know that $i \in \mathcal{C}(1, r_1)$, and so item i was added to the order at s at the moment it was temporarily opened. Thus, (i,t) was frozen at $open(r_1)$ or perhaps earlier. This implies that $open(r_1) \leq freeze(i,t)$. We also know that $R' \cap [open(r_1), r_1) = \emptyset$ (since we permanently opened r_1) and that $[freeze(i,t), r_1) \subseteq [open(r_1), r_1)$. This concludes the proof of the lemma for this case.

Case 2. The order of item i in period r_1 is an extra order. We know that the extra order at r_1 has some initiator (j^*, t^*), where $j^* \in \mathcal{S}(i)$ is the initiator item. Consider $\mathsf{N}_{j^*}(r_1)$, the freezing order of (j^*, t^*). In particular, we have already seen that $r_1 \leq \mathsf{N}_{j^*}(r_1) \leq t$. We claim that $freeze(j^*, t^*) \leq freeze(i,t)$. Observe that (j^*, t^*) was frozen when item j^* was added to the order at $\mathsf{N}_{j^*}(r_1)$. However, since $i \in \mathcal{C}(j^*, \mathsf{N}_{j^*}(r_1))$, it follows that item i was added to the order at $\mathsf{N}_{j^*}(r_1)$ together with item j^*. Thus, (i,t) was frozen together with (j^*, t^*) or perhaps earlier, so indeed $freeze(j^*, t^*) \leq freeze(i,t)$. By the construction

of the algorithm, we know that there does not exist an order with item j^* in $R' \cap [freeze(j^*, t^*), r_1)$. Since the solution is nested (i.e., if we order item i, we must also order item j^*), there does not exist any order with item i in $R' \cap [freeze(j^*, t^*), r_1)$. Since we have already concluded that $[freeze(i,t), r_1) \subseteq [freeze(j^*, t^*), r_1)$, we see that the lemma holds. ■

Corollary 5.2 *Any demand point (i,t) can contribute towards at most two orders in R'.*

Proof : Suppose that (i,t) contributes towards more than one order in R', and let $r_1 > r_2$ be the two latest such orders. We will show that it can not be the case that $r_1 < freeze(i,t)$. The rest of the proof is identical to that of Corollary 4.2.

Suppose that indeed $r_1 < freeze(i,t)$; in that case, the orders of item i at r_1 and r_2 must both be extra orders (since they do not lie in the active interval of (i,t)). Let $j^* \in \mathcal{P}(i)$ be the initiator item of the order at r_2 and let $\mathsf{N}_{j^*}(r_2)$ be the freezing order of the initiator (j^*, t^*). To show a contradiction it's sufficient to show that $t^* < r_1$ (since we must also have $r_1 < freeze(i,t) < \mathsf{N}_{j^*}(r_2)$). Recall that since the solution is nested, we have included all of the items $j \in \mathcal{S}(i)$ in the order at r_1 (either as a regular or as an extra order), including item j^*. Since $freeze(j^*, t^*) \leq r_2 < r_1$, we must have that $t^* < r_1$ (since otherwise (t^*, j^*) could not have been an initiator). We now complete the proof exactly along the lines of Corollary 4.2. ■

Corollary 5.3 *Consider a demand point (i,t) and let r_1 be the latest order towards which (i,t) contributes some positive share. Then the holding cost that (i,t) incurs in $(\hat{x}, \hat{y})$ is at most $\bar{H}^i_{r_1,t}$ (i.e., $\hat{H}^i_t \leq \bar{H}^i_{r_1,t}$).*

Proof : Same as in Corollary 4.3. ■

Theorem 5.4 *The primal-dual framework provides a 2-approximation algorithm to the assembly problem.*

Proof : Consider any demand point (i,t) and let $r_1 \in R'$ again be the latest order in $(\hat{x}, \hat{y})$ towards which (i,t) contributes a positive share. If the order of item i in period r_1 is a regular order, then (i,t) contributes $\sum_{j \in \mathcal{S}(i)} \hat{z}^{ij}_{r_1,t} > 0$. If the order of item i at r_1 is an extra order, then (i,t) contributes $\sum_{j \in path_{i,j^*}} \hat{z}^{ij}_{\mathsf{N}_{j^*}(r_1),t} > 0$, where $j^* \in \mathcal{S}(i)$ is the corresponding initiator item, and $freeze(i,t) \leq \mathsf{N}_{j^*}(r_1) \leq t$. In either case, this is clearly bounded by $\hat{b}^i_t$.

Now assume that (i,t) also contributes towards a second (earlier) order r_2. By Lemma 5.1, the order of item i at r_2 must be an extra order, such that $r_2 \notin [freeze(i,t), t]$. If $j' \in \mathcal{S}(i)$ is the corresponding initiator item of this order, then (i,t) contributes $\sum_{j \in path_{ij'}} \hat{z}^{ij}_{\mathsf{N}_{j'}(r_2),t} > 0$ towards r_2, and $freeze(i,t) \leq \mathsf{N}_{j'}(r_2) < r_1$ (see Corollary 5.2). We shall argue that:

$$\hat{b}^i_t = \bar{H}^i_{\mathsf{N}_{j'}(r_2),t} + \sum_{j \in \mathcal{S}(i)} \hat{z}^{ij}_{\mathsf{N}_{j'}(r_2),t} \geq \bar{H}^i_{\mathsf{N}_{j'}(r_2),t} + \sum_{j \in path_{ij'}} \hat{z}^{ij}_{\mathsf{N}_{j'}(r_2),t} \geq \bar{H}^i_{r_1,t} + \sum_{j \in path_{ij'}} \hat{z}^{ij}_{\mathsf{N}_{j'}(r_2),t}.$$

The first inequality follows from $\hat{z}^{ij}_{\mathsf{N}_{j'}(r_2),t} \geq 0$ ($\forall j \in \mathcal{S}(i)$), and the second inequality follows from the monotonicity of the holding costs and $\mathsf{N}_{j'}(r_2) < r_1$. From Corollary 5.3, we get that $\hat{b}^i_t \geq \hat{H}^i_t + \sum_{j \in path_{ij'}} \hat{z}^{ij}_{\mathsf{N}_{j'}(r_2),t}$. Corollary 5.2 and the fact that each demand point can contribute only towards orders $r \in R'$ with item i also imply that (i,t) does not contribute towards any order $r \in R'$ other than r_1 and r_2. As a result, we get that the sum of the holding cost incurred by (i,t) and its contributions towards ordering costs is bounded by $2\hat{b}^i_t$. This proves the theorem. ■

We note that the analysis will go through even if we allow the item ordering cost parameter of item 1 (K_1) to vary arbitrarily over time. We can also allow the item ordering cost of each item $i > 1$ to be a

non-decreasing function of the ordering time.

We end the discussion on the assembly problem by mentioning that under our general assumptions on the cost parameters, the variant of the assembly problem we consider is NP-Hard. This can be shown by a simple reduction from the JRP to the 2-stage assembly problem. Given an instance of the JRP, we rescale the demand and the holding cost parameters h^i_{st} of the items (by inversely proportionate value) so that for each period t ($t = 1, .., T$), there is a uniform demand $D_t = d_{it}$. Each of the items is the predecessor of a common dummy item 0 with ordering cost equal to the joint ordering cost K_0, demand D_t, and echelon holding cost equal to 0. This yields an instance of a 2-stage assembly problem, and since we can restrict to nested policies, it is equivalent to the original JRP instance.

6 Conclusions

In this paper we have shown a general algorithmic framework of how to generate optimal and near-optimal solutions to a class of classical deterministic inventory models.

Although the method is based on LP relaxations, our approximation algorithms do not require the LP's to be solved. They are used only in the analysis of the algorithms. The algorithms are clearly polynomial-time but there is still work to do so as to get the most efficient implementations. We believe that it would be interesting to test the typical quality of the solutions that our algorithms generate on different inputs and compare them to other known heuristics.

A very interesting theoretical open question is related to the approximability of the JRP. The problem is NP-hard but we know of no approximability hardness result and one can not even exclude the existence of a polynomial-time approximation scheme (i.e., one might be able to design a $\rho-$approximation algorithm for any $\rho > 1$). We mention again that for the assembly network problem with the traditional holding cost structure, it is not known whether it is NP-hard. A more specific open question is related to the tightness of the LP relaxations considered in this paper. We have constructed [22] an example in which the integrality gap is 1.21. This implies that using the LP as the only lower bound, one can not hope to prove a performance guarantee better than 1.21. However, there still exists a significant gap between the upper bound of 2 and the lower bound of 1.21.

Acknowledgments We thank Chaitanya Swamy for stimulating discussions and helpful suggestions.

References

[1] A. Archer. Inapproximability of the asymmetric facility location and k-median problems. Unpublished manuscript, 2000.

[2] E. Arkin, D. Joneja, and R. Roundy. Computational complexity of uncapacitated multi-echelon production planning problems. *Operations Research Letters*, 8:61–66, 1989.

[3] Y. Askoy and S. S. Erenguk. Multi-item inventory models with coordinated replenishment: a survey. *International Journal of Operations and Production Management*, 8:63–73, 1988.

[4] I. Bárány, T. J. Van Roy, and L. A. Wolsey. Uncapacitated lot-sizing: the convex hull of solutions. *Mathematical Programming Study*, 22:32–43, 1984.

[5] D. Bertsimas, C. Teo, and R. Vohra. On dependent randomized rounding algorithms. *Operations Research Letters*, 25:105–114, 1999.

[6] M. R. Bussieck, A. Fink, and M. E. Lübbecke. Yet another note on "An efficient zero-one formulation of the multilevel lot-sizing problem". Technical report, Department of Mathematical Optimization, Braunschweig University of Technology, 1998.

[7] J. Chuzhoy, S. Guha, S. Khanna, and J. Naor. Asymmetric k-center is log*n-hard to approximate. *Electronic Colloquium on Computational Complexity (ECCC) (038)*, 2003.

[8] W. B. Crowston and M. H. Wagner. Dynamic lot size models for multi-stage assembly systems. *Management Science*, 20:14–21, 1973.

[9] A. Federgrun and M. Tzur. The joint replenishment problem with time-varying parameters: efficient, asymptotic and epsilon-optimal solutions. *Operations Research*, 42:1067–1087, 1994.

[10] M. X. Goemans and D. P. Williamson. A general approximation technique for constrained forest problems. *SIAM Journal on Computing*, 24:296–317, 1995.

[11] E. Halperin, G. Kortsarz, and R. Krauthgamer. Tight lower bounds for the asymmetric k-center problem. *Electronic Colloquium on Computational Complexity (ECCC) 10(035)*, 2003.

[12] K. Jain and V. V. Vazirani. Approximation algorithms for metric facility location and k-median problems using the primal-dual schema and Lagrangian relaxation. *Journal of the ACM 48*, pages 274–296, 2001.

[13] D. Joneja. Multi-echelon and joint replenishment production and distribution systems with non stationary demand. Technical Report 731, School of OR&IE, Cornell University, 1987.

[14] D. Joneja. *Planning for joint replenishment and assembly systems with deterministic non-stationary demands*. PhD thesis, School of OR&IE, Cornell University, Ithaca, NY, January 1989.

[15] D. Joneja. The joint replenishment problem: new heuristics and worst case performance bounds. *Operations Research*, 38:723–771, 1990.

[16] E. Kao. A multi product dynamic lot size model with individual and joint setup costs. *Operations Research*, 27:279–289, 1979.

[17] J. Krarup and O. Bilde. Plant location, set covering and economic lot sizing: an O(mn) algorithm for structural problems. In *Numerische Methoden Bei Optimierungsaufgaben*, volume 3, pages 155–180, 1977.

[18] R. Levi and R. O. Roundy. A note on Joneja's joint replenishment problem approximation algorithm. In preparation.

[19] P. Raghavan and M. R. Rao. The multi-item lot sizing problem with joint replenishment: a polyhedral approach. Technical Report SOR-91-8, Stern School, NY University, 1991.

[20] P. Raghavan and M. R. Rao. Formulations to the multi-item lot sizing problem with joint replenishment. Technical Report SOR-92-19, Stern School, NY University, 1992.

[21] R. O. Roundy. Efficient, effective lot-sizing for multi-product, multi-stage production systems. *Operations Research*, 41:371–386, 1993.

[22] R. O. Roundy, R. Levi, and D. Shmoys. A lower bound on the integrality gap for a strong IP formulation for the joint replenishment problem. In preparation.

[23] Z. J. Shen, D. Simchi-Levi, and C. P. Teo. Approximation algorithms for the single-warehouse multi-retailer problem with piecewise linear cost structures. url: citeseer.nj.nec.com/439759.html.

[24] D. B. Shmoys, C. Swamy, and R. Levi. Facility location with service installation costs. To appear in *Proceedings of the 15th Annual SIAM-ACM Symposium on Discrete Algorithms*, 2004.

[25] D. Simchi-Levi, 2002. Private communication.

[26] A. Veinott. Minimum concave cost solutions of Leontief substitution models of multi-facility inventory systems. *Operations Research*, 17:262–291, 1969.

[27] H. M. Wagner and T. M. Whitin. Dynamic version of the economic lot sizing model. *Management Science*, 5:89–96, 1958.

[28] W. I. Zangwill. A deterministic multi-product, multi-facility production and inventory model. *Operations Research*, 14:486–507, 1966.