MATHEMATICS OF OPERATIONS RESEARCH
Vol. 31, No. 2, February 2006, pp. 267–284
ISSN 0364-765X | EISSN 1526-5471 | 06 | 3102 | 0267

DOI 10.1287/moor.1050.0178

Primal-Dual Algorithms for Deterministic Inventory Problems

Retsef Levi
IBM T. J. Watson Research Center, P.O. Box 218, Yorktown Heights, New York 10598, retsef@us.ibm.com

Robin O. Roundy
School of Operations Research and Industrial Engineering, Cornell University, Ithaca, New York 14853, robin@orie.cornell.edu

David B. Shmoys
School of Operations Research and Department of Computer Sciences, Cornell University, Ithaca, New York 14853, shmoys@cs.cornell.edu

We consider several classical models in deterministic inventory theory: the single-item lot-sizing problem, the joint replenishment problem, and the multistage 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 nonempty 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 applications in other settings. More specifically, we provide 2-approximation algorithms for the joint replenishment problem and for 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 byproduct of our work, we prove known and new upper bounds on the integrality gap of some linear-programming (LP) relaxations of the abovementioned problems.

Key words: approximation algorithms; primal-dual algorithms; inventory models
MSC2000 subject classification: Primary: 90B05; secondary: 68W25
OR/MS subject classification: Primary: inventory/production, approximations/heuristics; secondary: production/scheduling, approximations/heuristics
History: Received February 27, 2004; revised January 31, 2005, and August 16, 2005.

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 *multistage 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., Askoy and Erenguk [3], Joneja [12]). 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.") Because 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 nonnegative and, for each (i, t), is a nonincreasing 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 $(d_1, \ldots, d_T)$. 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 nonnegative 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 $(d_1, \ldots, d_T)$. 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 $\sum_{l=s}^{t-1} h^i_l$ in a 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 §4.2, 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 in §§3 and 4. 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 JRP. 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 JRP is NP-hard (Arkin et al. [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 (Zangwill [27], Veinott [25], Kao [15]) (by fixing the times at which joint orders are placed, the problem decomposes to independent single-item problems). 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 et al. [22], 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 LP relaxations, i.e., that provide tight lower bounds (see, e.g., Joneja [13], Raghaven and Rao [18, 19]). 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 [14], 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 (Simchi-Levi 2002, private communication). A somewhat different analysis yields a performance guarantee of 5 (Levy and Roundy [17]). Federgruen and Tzur [9] proposed an interesting dynamic-programming-based heuristic for the JRP, 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 and Within [26]. Furthermore, Krarup and Bilde [16] showed, in this case, that the facility location-inspired LP has integer optima by means of a primal-dual algorithm. Bárány et al. [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 et al. [5] gave a proof that 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, because 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 property of this LP formulation. Another dual-

based optimization algorithm for the single-item lot-sizing problem has been proposed by van Hoesel et al. [24]. However, their algorithm is very different than ours.

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 [20], who gave a 2-approximation algorithm (again for the case where all cost parameters are fixed over time) using a nonlinear 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 JRP. However, for the variant of the problem considered by Roundy [20], it is still not known to be NP-hard (Bussieck et al. [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 Roundy et al. [21], 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 correspond 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., Archer [1], Chuzhoy et al. [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 multicommodity models; there has been recent work that considers multicommodity facility location problems but, of course, with a symmetric cost metric (Shmoys et al. [23]).

We note that our algorithms have their intellectual roots in the seminal paper of Jain and Vazirani [11], 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 this paper is organized as follows. In §2, we describe the generic primal-dual algorithm focusing on the JRP case. In §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 §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 backorders are allowed. In §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 begin 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, because 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 and Vazirani [11] for the facility location problem (or that of Goemans and 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 continuous variable that indicates the current location of the *wavefront*; initially, $\tau = T$. The budget of any unfrozen demand point is then related to the wavefront location 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 so $\tau \leq t$) and as long as the

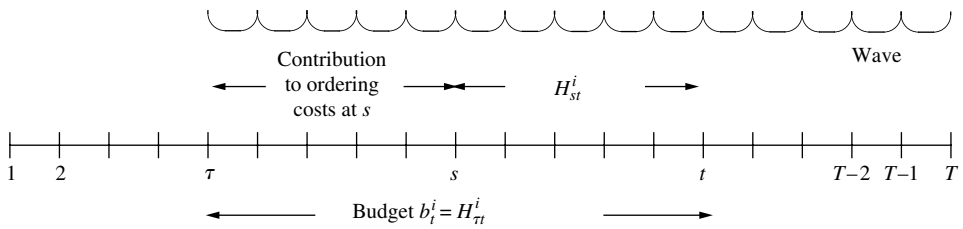

FIGURE 1. The waveform specification of the budget b^i_t and its allocation.

budget b^i_t is not frozen, we keep the budget of (i, t) equal to the holding cost of providing d_{it} from τ; that is, $b^i_t = d_{it} \cdot h^i_{\tau t}$, which, for notational convenience, we denote by $H^i_{\tau t}$ (see Figure 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 at time period s ($s = 1, \ldots, t$), the budget b^i_t of demand point (i, t) 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 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 all time periods s or later for those items ordered in time period s. We note that the waveform mechanism ensures that the budget of any frozen demand point is, at minimum, 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, because 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 each temporarily opened joint order in some period s, we consider the location τ of the wavefront when s was temporarily opened; let open(s) denote this value, where it is clear that open$(s) < s$. We then say that two orders s and r are *dependent* if and only if the shadow intervals $[\text{open}(s), s]$ and $[\text{open}(r), r]$ intersect. Observe that once a joint order is temporarily opened, we freeze the budgets of all unfrozen demand points that have paid a share toward the joint ordering cost of this order (a demand point can pay a share toward the joint ordering cost only if the item ordering cost is already fully paid). This implies that the joint ordering cost of each two orders with disjoint shadow intervals is paid by budgets of distinct demand points, i.e., the budget of each demand point is used to pay the joint ordering cost of at most one order. Next, we consider the temporarily opened orders from earliest to latest, and *permanently open* an order s if and only if its associated shadow interval does not intersect with the shadow 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. We want to make sure that each demand point (i, t) is satisfied from a joint order that includes item i and such that the holding cost incurred can be paid by the budget b^i_t. This requires additional work. Moreover, to achieve this, we may incur payments from the same budget toward the item ordering cost of multiple orders in the pruned solution. As we will show in the coming sections, the extent of this overpayment will be bounded, and each demand point will have to pay a share of the item ordering cost of at most two orders.

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^i_t. We show that for the JRP, one can pay for the cost of the solution in such a way that no demand point is charged more than twice its budget b^i_t. 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:

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

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

$$x_{st}^i \le y_s^i, \quad i = 1,\dots,N,\ t = 1,\dots,T,\ s = 1,\dots,t, \tag{2}$$

$$x_{st}^i \le y_s^0, \quad i = 1,\dots,N,\ t = 1,\dots,T,\ s = 1,\dots,t, \tag{3}$$

$$x_{st}^i,\ y_s^i,\ y_s^0 \ge 0, \quad i = 1,\dots,N,\ s = 1,\dots,T,\ t = s,\dots,T. \tag{4}$$

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 whether *any* item was ordered in period s. Constraint (1) ensures that each demand point (i,t) is satisfied from some time period $s \le t$. Constraint (2) ensures that no demand for item i can be provided from period s without placing an order for item i at s. 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

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

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

$$\sum_{t=s}^{T} l_{st}^i \le K_i, \quad i = 1,\dots,N,\ s = 1,\dots,T. \tag{6}$$

$$\sum_{i=1}^{N}\sum_{t=s}^{T} z_{st}^i \le K_0, \quad s = 1,\dots,T, \tag{7}$$

$$l_{st}^i,\ z_{st}^i \ge 0, \quad i = 1,\dots,N,\ t = 1,\dots,T,\ s = 1,\dots,t. \tag{8}$$

Naturally, (D) provides a lower bound on the cost of any feasible solution to the JRP, because 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 $s = 1,\dots,t$) so as to be served by one of them. Each potential order $s = 1,\dots,T$ 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 satisfied 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) toward 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 definition to $\tau \in (s-1,s)$ 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 the wavefront arrives at s, i.e., $\tau = s$ (for $s = T, T-1, \ldots, 1$), we identify all unfrozen demand points (i,t) with $t = s, \ldots, T$ and start increasing the variable l^i_{st} at the same rate as b^i_t. In other words, as long as b^i_t is not frozen (and $\tau \le t$), we keep $b^i_t = H^i_{\tau t} = H^i_{st} + l^i_{st}$; the variable l^i_{st} is the portion in the budget b^i_t that is used to pay for a share of the item ordering cost of item i in time period s. Because the wavefront moves backward in time, from now on we have $\tau \le s$. Moreover, because of the monotonicity of H, we know that $b^i_t = H^i_{\tau t} \ge H^i_{st}$, and from now on (i,t) will pay some share $l^i_{st} = H^i_{\tau t} - H^i_{st} \ge 0$ of the item ordering cost in s. (Note that as the wavefront reaches s and the budget b^i_t increases to H^i_{st}, constraint (5) becomes tight. As the budget increases further as the wavefront "advances" from s toward $s-1$, 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 $\sum_{t \ge s} l^i_{st} = K_i$. (Note that this means that we can no longer increase any variable l^i_{st} without violating 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 $t \ge s$. We freeze the variables l^i_{st} and instead start increasing the variables z^i_{st} (at the same rate as the budget b^i_t). We then have that $b^i_t = H^i_{\tau t} = H^i_{st} + l^i_{st} + z^i_{st}$, where z^i_{st} accounts for the portion in the budget b^i_t 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 $t \ge s$.

Event 3. Suppose that for some period $s > 1$, $\sum_{i=1}^N \sum_{t \ge s} z^i_{st} = K_0$. (Note that we can no longer increase any variable z^i_{st} without violating 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 $\sum_{t \ge s} l^i_{st} = K_i$. For each such item i, we freeze the budget of any unfrozen demand point (i,t) with $t \ge s$ (see also Event 2, above).

Event 4. Suppose that $\tau = 1$. We then open a joint order in period 1. We add to this order all the items $i = 1, \ldots, N$. We then charge the cost of this order to the dual variables of the demand points $(i,1)$ by setting $b^i_1 := l^i_{11} + z^i_{11}$, where $l^i_{11} := K_i$ and $z^i_{11} := K_0/N$ (for $i = 1, \ldots, N$). (Observe that if $\sum_{i=1}^N d_{i1} > 0$, then any feasible solution will place a joint order in period 1 and this order will include any item i with $d_{i1} > 0$.) Next, we freeze all the unfrozen budgets and terminate.

We note that the various events described above are likely to occur at noninteger wavefront locations (i.e., for noninteger 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 is a feasible dual solution. Moreover, the above procedure also induces a feasible (integer) primal solution, because the budget of each demand point is frozen only when there is a temporarily opened order s (such that $s \le t$) with item i. However, this solution is rather expensive, because the budget of a demand point can be multiply used to pay toward 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. More specifically, for the JRP and the assembly problem, we will show that each demand point contributes to at most two orders in the pruned solution. We first discuss the simpler special case of the single-item lot-sizing problem (in §3) and then discuss the more general model of the JRP (in §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, because 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:

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

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

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

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

We also get similar dual:

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

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

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

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

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 §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 (note that open$(s_1) = 1$)). We say that the interval $[\text{open}(s), s]$ 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 also say that a demand point t *contributes* toward the ordering cost of the order in period s if $\hat{l}_{st} > 0$. Observe that if the interiors of the shadow intervals of two orders, r and s, intersect, then there exists at least one demand point t that contributes toward both s and r (if $s < r$, then demand point r is such a point). Conversely, each two orders $s < r$ whose shadow intervals have disjoint interiors do not share any demand point that contributes toward their ordering cost, because all the budgets of demand points $t \ge r$ were frozen no later than open(r)). This property will be used in the proof of Lemma 3.1. We now 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. For an illustration of the pruning phase, see Figure 2. 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 point from the latest possible order in 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 each 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 $\sum_{t \ge s} \hat{l}_{st} = K$. Recall that a demand point t contributes toward an order $s \in R'$ if $\hat{l}_{st} > 0$. In addition, each demand point should pay for its holding cost. We use $\widehat{H}_t$ 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., $\hat{b}_t = H_{\text{freeze}(t), t}$. We call the interval $[\text{freeze}(t), t]$ 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 toward temporarily opened orders within its active interval. For each order s outside the interval, the holding cost H_{st} is higher than the budget $\hat{b}_t = H_{\text{freeze}(t), t}$, so no share can be paid toward the ordering cost.

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

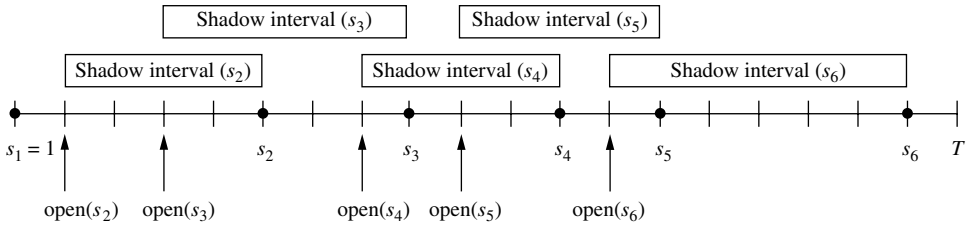

FIGURE 2. Pruning stage: Permanently open s_1, s_2, s_4, s_6, and so on.

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 $\text{open}(s') = \text{freeze}(t)$. If $s' \in R'$, then because 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 $\text{freeze}(t) = \text{open}(s') \le s < s' \le t$. However, this implies that $s \in [\text{freeze}(t), t]$, i.e., s is in the active interval of t.

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

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

THEOREM 3.1. *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 toward exactly one order $s \in R'$. More specifically, the share that demand point t contributes toward 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 generalize the input to allow the cost of placing an order in period s to be the time-dependent parameter K_s, the identical algorithm and analysis yield the same theorem in this more general setting.

4. The joint replenishment problem. We now describe the second phase of the primal-dual algorithm for the JRP and give its analysis. The pruning phase for the JRP is more involved than it is for the lot-sizing problem, because 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 $\text{open}(s)$ (for orders $s \in R$), and $\text{freeze}(i, t)$ (for each demand point (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 some demand point (i, t) with $t \ge s$ pays a share of the joint ordering cost at s (i.e., $\sum_{t \ge s} \hat{z}^i_{st} > 0$). Let $\mathcal{C}(s)$ be the set of contributor items for each $s \in R$. Observe that each of the items in $\mathcal{C}(s)$ was added to the joint order at s immediately when this order was temporarily open in $\text{open}(s)$ as part of Event 2(b) above.

We start by applying the same procedure we used in §3 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'). As before, after this pruning step, each demand point (i, t) contributes toward the ordering cost of at most one joint order in R' (the arguments are identical to the ones in Lemma 3.1). Initially, for each 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 joint order within the active interval of (i, t) that includes item i. Thus, we cannot guarantee that each demand point incurs bounded holding cost, and more work is required. We will apply a "correction step" to make sure that each demand point (i, t) will be satisfied by an order within its active interval. This will imply that the holding cost it incurs can be paid by its budget $\hat{b}^i_t$. This step is done separately for each item i.

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 in $R' \cap [\text{freeze}(i, t), t]$, say s. We shall say that (i, t) is the *initiator* of the extra order of item i in period s. That is, we indicate that item i was added to s as an extra order to provide (i, t) with a "close" order. This process is repeated on the remaining time horizon $[1, s)$, where again s is the earliest extra order of item i we have placed so far. Similarly, we next search for the latest positive demand point (i, t) with $t \in [1, s-1]$ and with no appropriate order (with item i) in its active interval. We continue until each demand point (i, t) can be served, by either a regular or an extra order, within its active interval (see Figure 3). The same procedure is repeated for

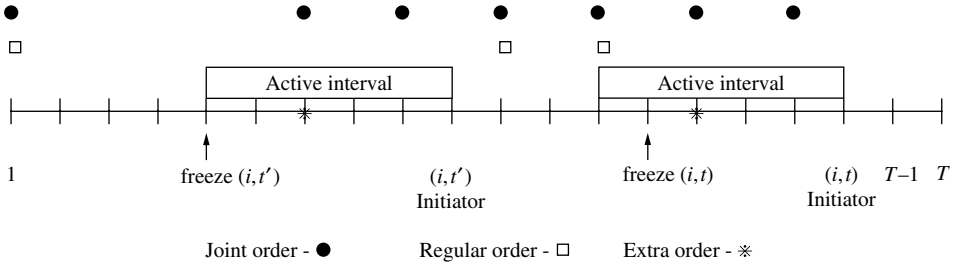

FIGURE 3. Correction step for item i: (i, t) is the latest positive demand point with no regular order of item i (square) in its active interval, so add an extra order (star) of item i in the earliest permanently opened joint order (bullet) within its active interval. Then, (i, t') is the next demand point with no regular order in its active interval, repeat, and add another extra order.

each item i. Unlike the lot-sizing case discussed in §3, the extra orders that we add in the JRP case are likely to create overpayment, where the budgets of some demand points may be used to pay a share of the item ordering cost of multiple orders. However, by choosing the extra orders to be the earliest possible in the active intervals of the corresponding initiators, we guarantee that each demand point can pay toward at most two orders. This will be made rigorous in the analysis presented below.

We note that after all of the orders are specified, each demand point (i, t) is satisfied from the latest possible order $s \in R'$ containing item i, and this provides a feasible solution for the JRP. Let $(\hat{x}, \hat{y})$ denote this solution.

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 $2\hat{b}_t^i$. For this, we next introduce a somewhat more involved charging scheme that specifies how each demand point (i, t) is charged.

For the joint and regular orders, we use the contributor items to pay for their joint and item ordering cost, respectively. The joint ordering cost and the item ordering cost of regular orders of each $s \in R'$ is

$$K_0 + \sum_{i \in \mathscr{C}(s)} K_i = \sum_{i \in \mathscr{C}(s)} \sum_{t \geq s} (\hat{l}_{st}^i + \hat{z}_{st}^i).$$

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), i.e., the order that caused the budget of (i, t) to freeze. In particular, $\text{freeze}(i, t) \leq \text{open}(s')$ (observe that the budget $\hat{b}_t^i$ could have frozen after s' was opened as described in Event 2(b) in §2). We claim that $s \leq s'$. By definition, s is the earliest order in $R' \cap [\text{freeze}(i, t), t]$. We also know that $R' \cap [\text{open}(s'), s'] \neq \emptyset$, because either $s' \in R'$ (i.e., s' is permanently open), or it was eliminated by some earlier order $s'' \in R'$, such that $\text{open}(s') \leq s'' < s'$. Consequently, indeed $s \leq s'$. Because s' is the freezing order of demand point (i, t), it follows that $\sum_{t' \geq s'} \hat{l}_{s',t'}^i = K_i$; we can use this to pay for the cost of the extra order of item i at s. We essentially use the item ordering cost of the order in s' that may have not been permanently opened to pay the item ordering cost of the extra order of 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. We note that each extra order is paid by a (possibly unopened) order $\mathsf{N}_i(s)$ later in time, i.e., if we placed an extra order of item i in period s, then $s \leq \mathsf{N}_i(s)$. However, it is possible that $s = \mathsf{N}_i(s)$ if (i, t) was frozen by s, and item i was not a contributor of s, and then was added to s as an extra order.

Consider any demand point (i, t); we will say that (i, t) *contributes toward some regular order* $s \in R'$ if $i \in \mathscr{C}(s)$ and $\hat{z}_{st}^i + \hat{l}_{st}^i > 0$. In addition, we will say that (i, t) *contributes toward some extra order* of item i in period $s \in R'$ if $\hat{l}_{\mathsf{N}_i(s),t}^i > 0$. Observe that each demand point (i, t) can only contribute toward a joint and a regular order r with $r \leq t$ and toward an extra order s with $\mathsf{N}_i(s) \leq t$. We charge demand point (i, t) with what it contributes toward different orders in R', as well as the holding cost it incurs in $(\hat{x}, \hat{y})$. Denote this holding cost by $\hat{H}_t^i$.

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 $2\hat{b}_t^i$. The main idea will be to show that each demand point (i, t) cannot contribute to more than two orders (see Lemma 4.1 and Corollary 4.1 below). This property implies that the contribution of each demand point (i, t) toward the latest order to which it contributes is bounded by $\hat{b}_t^i$, and that the holding cost it incurs plus the possible contribution toward a second order are also bounded by $\hat{b}_t^i$.

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, toward which (i, t) contributes. Then, either $r_1 \notin [\text{freeze}(i, t), t]$ or it is the earliest order in $R' \cap [\text{freeze}(i, t), t]$.*

PROOF. Assume that $r_1 \in [\text{freeze}(i,t), t]$ 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 $\text{open}(r_1) \le \text{freeze}(i,t)$. We know that $i \in \mathscr{C}(r_1)$, and so $\sum_{u \ge r_1} \hat{z}^i_{r_1,u} > 0$. 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 $\text{open}(r_1)$); 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, $\text{open}(r_1) \le \text{freeze}(i,t)$. By the choice of r_1, we know that its shadow interval $[\text{open}(r_1), r_1]$ does not contain another order $r \in R'$. Because $[\text{freeze}(i,t), r_1] \subseteq [\text{open}(r_1), r_1]$, this implies that r_1 is the earliest order in $R' \cap [\text{freeze}(i,t), t]$.

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 $\mathsf{N}_i(r_1)$. As we have already observed, we must have $r_1 \le \mathsf{N}_i(r_1) \le t$ (we assume that (i,t) contributes toward r_1). In particular, by the waveform properties, we know that $\text{freeze}(i,t^*) \le \text{freeze}(i,t)$, because (i,t) was frozen no later than (i,t^*) was (because $\mathsf{N}_i(r_1) \le t$). 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, $R' \cap [\text{freeze}(i,t^*), r_1) = \emptyset$. Given that we already concluded that $\text{freeze}(i,t^*) \le \text{freeze}(i,t)$, the lemma follows. □

The above lemma has several immediate corollaries:

COROLLARY 4.1. *Any demand point (i,t) can contribute toward at most two orders in R'.*

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

Suppose that $r_1 < \text{freeze}(i,t)$; in that case, r_1 and r_2 must both be extra orders of item i (because they do not lie in the active interval of (i,t)). 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 $r_1 < \text{freeze}(i,t) \le \mathsf{N}_i(r_2)$. However, 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, $r_1 \in [\text{freeze}(i,t), t]$. By Lemma 4.1, it follows that r_1 is the earliest permanent order in $[\text{freeze}(i,t), t] \cap R'$. 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. However, we have already seen that there is at most one such order (namely, r_2), which completes the proof. □

COROLLARY 4.2. *Consider a demand point (i,t) and let r_1 be the latest order toward 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., $\widehat{H}^i_t \le H^i_{r_1,t}$).*

PROOF. Because 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, because r_1 is either the earliest in $[\text{freeze}(i,t), t] \cap R'$ or $r_1 < \text{freeze}(i,t)$. □

We are now ready to prove the main theorem.

THEOREM 4.1. *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 $(\hat{x},\hat{y})$ toward 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 $\text{freeze}(i,t) \le \mathsf{N}_i(r_1) \le t$. In either case, this is clearly bounded by $\hat{b}^i_t$. Moreover, if (i,t) contributes only to one order, then by Corollary 4.2, $\widehat{H}^i_t \le \hat{b}^i_t$, and we can pay for its holding cost and overall contributions using at most $2\hat{b}^i_t$.

Now assume that (i,t) also contributes toward a second (earlier) order r_2. By Lemma 4.1, r_2 must be an extra order of item i such that $r_2 \notin [\text{freeze}(i,t), t]$. Hence, (i,t) contributes $\hat{l}^i_{\mathsf{N}_i(r_2),t} > 0$ toward r_2, where $\text{freeze}(i,t) \le \mathsf{N}_i(r_2) < r_1$. The latter inequality follows from the fact that because 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 $\hat{b}^i_t = H^i_{\mathsf{N}_i(r_2),t} + \hat{l}^i_{\mathsf{N}_i(r_2),t} + \hat{z}^i_{\mathsf{N}_i(r_2),t} \ge H^i_{\mathsf{N}_i(r_2),t} + \hat{l}^i_{\mathsf{N}_i(r_2),t} \ge H^i_{r_1,t} + \hat{l}^i_{\mathsf{N}_i(r_2),t}$; the first inequality follows from $\hat{z}^i_{\mathsf{N}_i(r_2),t} \ge 0$, and the second inequality follows from the monotonicity of the holding costs and $\mathsf{N}_i(r_2) < r_1$. From Corollary 4.2, we get that $\hat{b}^i_t \ge \widehat{H}^i_t + \hat{l}^i_{\mathsf{N}_i(r_2),t}$. Corollary 4.1 also implies that (i,t) does not contribute toward 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 toward ordering costs is bounded by $2\hat{b}^i_t$. This proves the theorem. □

It is worthwhile mentioning that it is unclear whether the above analysis is tight. In Roundy et al. [21], we show that the integrality gap of the LP relaxation (P) is at least 1.23. However, there is still a significant gap

between the current guarantee of 2 and the proven lower bound of 1.23. We also note that the above analysis remains valid if we allow the joint ordering cost (K_0) to be time dependent. Because the extra orders are always paid by orders later in time ($s \le \mathsf{N}_i(s)$), we can also allow time dependent item ordering costs provided that they are nondecreasing over time. This will imply that $K^i_s \le K^i_{\mathsf{N}_i(s)}$, and the analysis goes through. If we allow arbitrary cost parameters, then there exists a simple reduction from the set cover problem, and hence, one cannot 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 nonnegative, linear in d_{it}, and nondecreasing in $s \ge t$ 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 costs 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 $\min\{H^i_{s,t'}, B^i_{t,t'}\}$, i.e., $G^i_{st} = \sum_{t' \in [s,t)} \min\{H^i_{s,t'}, B^i_{t,t'}\}$.

We now let $\overline{H}^i_{st} := G^i_{s,t+1} - G^i_{st}$ for each $s < t$, and let $\overline{H}^i_{ss} := H^i_{ss} = B^i_{ss}$. The parameter $\overline{H}^i_{st}$ accounts for the difference in the overall holding and back order costs 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 $\overline{H}$ parameters are nonnegative. Using this, we consider the LP in §2 having $\overline{H}$ 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 $\overline{H}^i_{st}$ with it, because 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), $\overline{H}^i_{st}$ is nonincreasing 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 $\overline{H}$ parameters) for the JRP with back orders. Because 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 §3. We believe that this is the first primal-dual algorithm for this variant of the lot-sizing problem.

LEMMA 4.2. *Consider some demand point (i, t) and some $1 < s < t$. Then, $\overline{H}^i_{st} \le \overline{H}^i_{s-1,t}$.*

PROOF. First, observe that the above inequality is equivalent to the inequality $G^i_{st} + G^i_{s-1,t} + \ge G^i_{s,t+1} + G^i_{s-1,t}$. For each demand point (i, t') and some $s_1 \le t' < s_2$, 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, $\overline{H}^i_{s_1,s_2} = \sum_{t' \in [s_1,s_2)} \Delta^{it'}_{s_1,s_2}$. Focus now on some demand point (i, t') with $t' \in [s, t]$. By the monotonicity assumption, we know that $\Delta^{it'}_{st} \ge 0$. It is sufficient to show that $\Delta^{it'}_{st} \le \Delta^{it'}_{s-1,t}$. 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 $H^i_{st'} < B^i_{t+1,t'}$, because 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 case, $\Delta^{it'}_{s-1,t} \ge \Delta^{it'}_{s-1,t} = H^i_{st'} - B^i_{tt'}$. This completes the proof. □

COROLLARY 4.3. *The primal-dual algorithm provides a 2-approximation algorithm for the JRP with back orders.*

COROLLARY 4.4. *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 to apply the primal-dual method to the assembly problem. In the assembly problem, we are given N items, $i = 1, \ldots, N$, out of which the first item ($i = 1$) is the only end product that is facing external demand. All of the other items ($i = 2, \ldots, N$) are in fact subassemblies that are used to build the end product. More precisely, 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 $(d_1, \ldots, d_T)$. This tree defines the assembly relations between the different items. Each unit of item i is assembled from one unit of each of its direct predecessor items in the tree. This ratio of "one-to-one" is without loss of generality and can be achieved by redefining the unit of measure of the items if necessary. The tree structure implies that each item is used to directly assemble a single more complex item, as it has a single successor item in the tree. The goal is to satisfy all of the external demands for item 1, $d_1, \ldots, d_T$, on time with minimum cost. The cost again consists of a fixed ordering cost, K_i for each order of item i, and a linear item holding cost for carrying inventory of that item over periods, where the holding cost has a similar structure to the one discussed in §1. However, each order of item i can be satisfied only from the on-hand inventory of the items from which this item is assembled (i.e., its direct predecessor items in the tree, or an external supplier if this item corresponds to a leaf in the tree). Alternatively, an order of item i placed in period t can be used to satisfy orders of its successor item in period t and in subsequent periods. In other words, to provide the demand d_t of item 1 in period t, each one of the items in the tree must provide d_t units by time t. 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 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.

One well-known result on the assembly problem is the optimality of what is called the *class of nested policies* (see Crowston and Wagner [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, item $\sigma(i)$. 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. It is not hard to show that any nonnested policy can be converted to a nested policy with at most the same cost by differing orders of the item with the larger index (for details, see Crowston and Wagner [8]). In addition, it is readily verified that *zero inventory ordering* (ZIO) policies are optimal for the assembly problem. In ZIO policies, we place an order for item i only if the current on-hand inventory of this item is 0. The proof is again by virtue of converting any policy that does not follow the ZIO rule to one that does follow it and has at most the same cost (in this case, by decreasing the quantity of one order and increasing the quantity of another). These two properties imply that the assembly problem also has an optimal policy in which for each item, the units that are used to satisfy the demand in period t are provided by a single order of item i.

In multistage 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 so far (which is essentially the on-hand inventory of that item). 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 its successor items (i.e., $j \in \mathcal{S}(i)$). Observe that the echelon inventory level of each item i increases only when an order of item i is placed, and decreases only when demand occurs. In other words, it is dependent only on decisions made with respect to item i. Using the echelon inventory concept, we can define for each item i and $s \le t$, a per unit *echelon holding cost*, $\bar{h}^i_{st}$, for ordering a unit of item i in period s that will be used to satisfy the demand of the end product in period t. We will assume that the echelon holding cost parameters have the same monotonicity properties as discussed in §1.

We now discuss several comments on the relationship between echelon and conventional holding cost. If we also assume that the conventional holding cost is additive, i.e., $h^i_{st} = h^i_{st'} + h^i_{t't}$ for each i and $s \le t' \le t$ (this is essentially the traditional holding cost with time-dependent parameters), then there exists a straightforward cost transformation from conventional to echelon holding cost. Given the conventional holding cost parameters h^i_{st}, we define 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 assume that $\bar{h}^i_{st}$ is nonnegative, because otherwise there will be no physical inventory of the predecessor items (it will be cheaper to assemble them immediately). To see

why echelon and conventional holding costs lead to equivalent formulations of the model, consider the demand d_t of item 1 in period t, and let $s(i) \le t$ be the period in which item i ordered these d_t units ($i = 1, \ldots, N$). More specifically, these units were assembled into item i in period $s(i)$, and then assembled into item $\sigma(i)$ in period $s(\sigma(i))$. Therefore, the conventional holding cost that these units incur at stage i is equal to $h^i_{s(i),s(\sigma(i))}d_t$. However, by the additivity assumption, this is equal to $(h^i_{s(i),t} - h^i_{s(\sigma(i)),t})d_t$. So the overall conventional holding cost incurred is

$$d_t \sum_{i=1}^{N}(h^i_{s(i),t} - h^i_{s(\sigma(i)),t}) = d_t \sum_{i=1}^{N}\left(h^i_{s(i),t} - \sum_{k\in\mathcal{P}'(i)} h^k_{s(i),t}\right) = d_t \sum_{i=1}^{N} \bar{h}^i_{s(i),t}.$$

Observe again that each item incurs echelon holding costs that are a function of the decisions made only with respect to item i (at stage i). This decomposition of holding costs to items will be a key property in formulating the LP below. From now on, we will assume that the problem is given to us with echelon holding cost parameters $\bar{h}^i_{st}$ as discussed above.

Let $\bar{H}^i_{st} = \bar{h}^i_{st} d_t$ be the overall echelon holding cost of ordering the d_t units of item i required to satisfy the demand in period t from period s. Relying on concepts of ZIO policies and echelon inventories and holding costs, it is straightforward to construct a correct integer-program formulation that solves the assembly problem, and admits an LP relaxation similar to the one given in §2:

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

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

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

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

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 constraint (3). The binary variable x^i_{st} corresponds to the decision to order the units of item i that are used to satisfy the demand in period t in period s. The objective function coefficient of x^i_{st} is the corresponding echelon holding cost $\bar{H}^i_{st}$ associated with this decision. 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. It is straightforward to verify that the induced integer program finds a minimum-cost feasible nested policy, i.e., it finds an optimal solution for the assembly problem (because nested policies are optimal). Thus, the LP relaxation again provides a lower bound on the cost of each feasible policy. Note that in the above LP there are many redundant constraints (instead of constraint (16), it would be sufficient only to require for each (i, t) and $s \le t$, that $x^i_{st} \le y^i_s$ and $x^i_{st} \le y^{\sigma(i)}_s$). However, because we are not going to solve the LP, it does not have any impact. On the other hand, we get a "nicer" dual problem:

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

$$\text{subject to} \quad b^i_t \le \bar{H}^i_{st} + \sum_{j\in\mathcal{S}(i)} z^{ij}_{st}, \quad i = 1, \ldots, N,\ t = 1, \ldots, T,\ s = 1, \ldots, t. \tag{18}$$

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

$$z^{ij}_{st} \ge 0, \quad i = 1, \ldots, N,\ t = 1, \ldots, T,\ s = 1, \ldots, t, \tag{20}$$

$$j \in \mathcal{S}(i).$$

The intuition underlying the dual program is similar to the one in the JRP case. We again think of the variables b^i_t as budgets associated with demand points. However, now a demand point (i, t) corresponds to the d_t units of item i that are assembled into the d_t units of the end product (item 1) that satisfy the external demand in period t. The concepts of echelon inventory and echelon holding cost enable us to treat each such demand point separately. Because the solution is nested, whenever there is an order of item i, in the same period there is an order of each one of the items on the path between i and the root of the tree (i.e., $j \in \text{path}_{i1}$). Hence, each demand point (i, t) will first pay a share of the item ordering cost of item i toward potential orders in periods $\{s\colon s \le t\}$. However, it may also pay some share of the item ordering cost of items on the path between i and the root of the tree. The variables z^{ij}_{st} for $j \in \mathcal{S}(i)$ account for these contributions.

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 again use the waveform mechanism, i.e., the budget of each demand point (i,t) is kept at 0 as long as $\tau \geq t$. Once $t \geq \tau$ and as long as the budget is not frozen, we maintain the invariant $b_t^i = \bar{H}_{\tau t}^i$ (the echelon holding cost of providing d_t units of demand i in period t from τ). We note again 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 $\bar{H}_{st}^i + \sum_{j \in \mathcal{S}(i)} z_{st}^{ij}$. First, 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. Once the budget is sufficiently large to pay for the holding cost, and because of the nestedness property, it will possibly contribute a share of the item ordering cost at s toward items $j \in \text{path}_{i1}$. The allocation of these contributions toward item ordering costs is done according to the order of these items on the path from i to 1, i.e., (i,t) may contribute to the item ordering cost in period s of some item $j \neq i$ in $\mathcal{S}(i)$ only after the item ordering costs in period s of each of the items on the path from i to j is already fully paid. These contributions are done 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, $\sum_{t \geq s} \sum_{k \in \mathcal{P}(i)} z_{st}^{ki} \leq K_i$ (constraint (19) above). 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 $j \in \text{path}_{i1}$) 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 the wavefront arrives at s, i.e., $\tau = s$ (for $s = T, T-1, \ldots, 2$), we identify all unfrozen demand points (i,t) with $t = s, \ldots, T$, 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 $\sum_{k \in \mathcal{P}(i)} \sum_{t \geq s} z_{st}^{ki} = K_i$, i.e., the item ordering cost of i in this period is fully paid. (Note that this means that we can no longer continue to increase any of the variables z_{st}^{ki} without violating 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 toward the item ordering cost of item i at s, i.e., the set of items $\{k \in \mathcal{P}(i)\colon \sum_{t \geq s} z_{st}^{ki} > 0\}$. Note that all of these items have the property that each $j \in \text{path}_{ki}$ 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 each unfrozen demand point (k,t) with $t \geq s$.

(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 toward the item ordering cost of item i at s will now start to contribute toward the item ordering cost of that item j' at s. More precisely, let $j' := \max\{j \in \mathcal{S}(i)\colon \sum_{k \in \mathcal{P}(j)} \sum_{t \geq s} z_{st}^{kj} < K_j\}$; clearly, $1 \leq j' < i$. Then, for each item $k \in \mathcal{P}(i)$ with $\sum_{t \geq s} z_{st}^{ki} > 0$, consider each unfrozen demand point (k,t) with $t \geq s$: 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 toward 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 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 each 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 that $\tau = 1$. We then open the order in period 1. We add to this order all of the items $i = 1, \ldots, N$. 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 $i = 1, \ldots, N$). 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). In addition, the induced primal solution is feasible for the assembly problem and is nested. However, this initial primal solution for the assembly problem is again potentially too expensive because of possible multiple payments, so we need to prune it.

5.3. The pruning phase. Our goal in the pruning step for the assembly problem is to find a cheaper feasible solution which is still nested. To achieve a nested feasible solution, we exploit the tree structure and perform the pruning phase in an iterative way, starting at item 1 and then going up the tree. We treat item i only when all of the permanent orders of its (single) successor items are already determined. Because we wish to keep the solution nested, the permanently open orders of item i will be a subset of the permanently open orders of its direct successor item $\sigma(i)$ that are assumed to be already determined. Let $R := \{s_1 = 1 < s_2 < \cdots < s_m\}$ be the set

of the time periods of all temporarily opened orders at the end of the first phase. Observe that each temporarily open order includes item 1, and possibly items on paths coming into item 1. To facilitate the presentation and the analysis of the pruning phase in 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 there exists some demand point (k, t) with $t \geq s$ and such that $\hat{z}_{st}^{ki} > 0$, or in other words, $\sum_{t \geq s} \hat{z}_{st}^{ki} > 0$. We will denote the set of contributor items by $\mathcal{C}(i, s)$. The following property of contributor items, which follows from the construction of the algorithm, plays a critical role in the execution of the pruning phase (to be described below) and the analysis of the algorithm. If item k is a contributor of an order of item i in period s, and item l is on the path from k to i, then item l is also a contributor of item i in period s. In other words, if $k \in \mathcal{C}(i, s)$, then $\text{path}_{ki} \subseteq \mathcal{C}(i, s)$.

We again use open(s) and the corresponding shadow interval (for each $s \in R$), and freeze(i, t) and the corresponding active interval (for each (i, t)). Here, open(s) is the wavefront location when the item ordering cost of item 1 at s was fully paid, and freeze(i, t) is the wavefront location at the time when the budget $\hat{b}_t^i$ was frozen, i.e., the first time when there was some period $s \leq t$ such that all the item ordering costs of items on path_{i1} were fully paid.

We start with item 1, and perform the same greedy procedure we used 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 $i = 2, \ldots, N$ in a way such that each item i is considered only after $\sigma(i)$ was considered. Focus now on some item $i > 1$. Before the pruning step of this item starts, we assume that we have already permanently opened all the orders of its successor item $\sigma(i)$. Moreover, some of these orders may include item i, exactly those orders for which item i was a contributor item for $\sigma(i)$ or some other item $j \in \mathcal{S}(i)$. As in the JRP case, there is no guarantee that for each demand point (i, t), there already exists a permanently opened order that includes item i within its active interval. We again have to perform a "correction step" similar to the one described for the JRP in §4. We start at the end of the planning horizon, i.e., at T, and look for the latest positive demand point, say (i, t), such that currently there does not exist a permanently opened order of item i within its active interval, $[\text{freeze}(i, t), t]$. Let $s' \in R$ be its freezing order. We now consider the earliest permanently opened order in $R' \cap [\text{freeze}(i, t), t]$ with item $\sigma(i)$, say s, and add to this order all of the contributor items of the order of i at s', i.e., items $k \in \mathcal{C}(i, s')$. Observe that all of these items are parents of item i, and were not yet considered in the pruning phase. We consider only permanently open orders that already include item $\sigma(i)$ to guarantee the nestedness of the solution. Recall that for each $k \in \mathcal{C}(i, s')$, it is also the case that each item l on the path from k to i (i.e., $l \in \text{path}_{ki}$) is also a contributor item (i.e., $l \in \mathcal{C}(i, s')$). These are called *extra orders*. We say that (i, t) and i are the *initiator* and the *initiator item*, respectively, of these extra orders in s. We note that in essence these extra orders are equivalent to the extra orders that we have added in the pruning phase for the JRP in §4. The only difference is that now we may add not only an extra order of item i, but a set of extra orders of a subset of parent items of i, namely, its contributor items at the freezing order s' (i.e., items in $\mathcal{C}(i, s')$). 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, i.e., that in the active interval of each demand point (i, t), there exists at least one order with item $\sigma(i)$. Moreover we argue that $s \leq s' = \mathsf{N}_i(s)$. Observe that for item i such that $\sigma(i) = 1$, the arguments are identical to the ones in the JRP case (see §4). So, for each i, we can assume by induction that the procedure is well defined for $\sigma(i)$. Let $s' \in R$ be again the freezing order of (i, t) and consider the demand point $(\sigma(i), s')$; we claim that $\text{freeze}(i, t) \leq \text{freeze}(\sigma(i), s')$. 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 in the process, i.e., $\text{freeze}(i, t) \leq \text{freeze}(\sigma(i), s')$. By induction, we know that when (i, t) is considered, we have already ensured that there exists a permanently open order in $R' \cap [\text{freeze}(\sigma(i), s'), s']$ with item $\sigma(i)$. Because $[\text{freeze}(\sigma(i), s'), s'] \subseteq [\text{freeze}(i, t), t]$, we conclude that the procedure described above is indeed well defined, and that $s \leq s'$.

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. The idea will be again to show that we can pay for the cost of the solution using the dual budgets, such that each demand point is charged at most $\hat{b}_t^i$.

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 each order $s \in R'$, the items included in this order can be

either regular orders or extra orders that were added in the course of the pruning phase. Let $I(s)$ be the set of the initiator items of the extra orders included in s in $(\hat{x}, \hat{y})$. In other words, the set of all items included in an order at $s \in R'$ is partitioned into $\mathscr{C}(1, s)$ and $\{\mathscr{C}(j, s)\colon j \in I(s)\}$.

We pay for the ordering cost of the regular orders at s, i.e., of items $i \in \mathscr{C}(1, s)$, using $\sum_{i\in\mathscr{C}(1,s)} K_i = \sum_{i\in\mathscr{C}(1,s)} \sum_{j\in\mathscr{S}(i)} \sum_{t\geq s} \hat{z}^{ij}_{st}$. The equality is correct based on the observation that if for some $k \in \mathscr{P}(i)$ and $j \in \mathscr{S}(i)$, we have $k \in \mathscr{C}(i, s)$ and $i \in \mathscr{C}(j, s)$, then we also have $k \in \mathscr{C}(j, s)$.

As for the items in extra orders at s, we can partition them according to their initiator item in $I(s)$. Thus, we have $\sum_{i\in I(s)} \sum_{k\in\mathscr{C}(i,\mathsf{N}_i(s))} K_k = \sum_{i\in I(s)} \sum_{k\in\mathscr{C}(i,\mathsf{N}_i(s))} \sum_{l\in\text{path}_{ki}} \sum_{t\geq\mathsf{N}_i(s)} \hat{z}^{kl}_{\mathsf{N}_i(s),t}$. 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 toward a regular order in period $s \in R'$ if $i \in \mathscr{C}(1, s)$ and $\sum_{j\in\mathscr{S}(i)} \hat{z}^{ij}_{st} > 0$. We say that (i, t) contributes toward extra orders at some $s \in R'$ if $i \in \mathscr{C}(j, \mathsf{N}_j(s))$ for some $1 < j \in I(s)$ and $\sum_{k\in\text{path}_{ij}} \hat{z}^{ik}_{\mathsf{N}_j(s),t} > 0$. In addition, each demand point is charged with the echelon holding cost that it incurs in $(\hat{x}, \hat{y})$; denote this cost by $\widehat{H}^i_t$. 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 $\hat{b}^i_t$.

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

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

PROOF. Assume $r_1 \in [\text{freeze}(i, t), t]$ 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 $i \in \mathscr{C}(1, r_1)$, and so item i was added to the order at s at the moment it was temporarily opened. Thus, (i, t) was frozen at $\text{open}(r_1)$ or perhaps earlier. This implies that $\text{open}(r_1) \leq \text{freeze}(i, t)$. We also know that $R' \cap [\text{open}(r_1), r_1) = \emptyset$ (because we permanently opened r_1) and that $[\text{freeze}(i, t), r_1] \subseteq [\text{open}(r_1), r_1]$. 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 $j^* \in \mathscr{S}(i)$ 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 $\text{freeze}(j^*, t^*) \leq \text{freeze}(i, t)$. Observe that (j^*, t^*) was frozen when item j^* was added to the order at $\mathsf{N}_{j^*}(r_1)$. However, because $i \in \mathscr{C}(j^*, \mathsf{N}_{j^*}(r_1))$, 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 $\text{freeze}(j^*, t^*) \leq \text{freeze}(i, t)$. By the construction of the algorithm, we know that there does not exist an order with item j^* in $R' \cap [\text{freeze}(j^*, t^*), r_1]$. Because 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 $R' \cap [\text{freeze}(j^*, t^*), r_1]$. Because we have already concluded that $[\text{freeze}(i, t), r_1] \subseteq [\text{freeze}(j^*, t^*), r_1]$, we see that the lemma holds. □

COROLLARY 5.1. *Each demand point (i, t) can contribute toward 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 cannot be the case that $r_1 < \text{freeze}(i, t)$. The rest of the proof is identical to that of Corollary 4.1.

Suppose that indeed $r_1 < \text{freeze}(i, t)$; in that case, the orders of item i at r_1 and r_2 must both be extra orders (because they do not lie in the active interval of (i, t)). Let $j^* \in \mathscr{S}(i)$ 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^*). Clearly, $\mathsf{N}_{j^*}(r_2) \leq t^*$, but we must also have $r_1 < \text{freeze}(i, t) < \mathsf{N}_{j^*}(r_2)$ (demand point (i, t) contributes toward the order of j^* at $\mathsf{N}_{j^*}(r_2)$). This implies that to show a contradiction, it is sufficient to show that $t^* < r_1 < \text{freeze}(i, t)$. Recall that because the solution is nested, we have included all of the items $j \in \mathscr{S}(i)$ in the order at r_1 (either as a regular or as an extra order), including item j^*. Because $\text{freeze}(j^*, t^*) \leq r_2 < r_1$ (demand point (j^*, t^*) is the initiator of the extra order at r_2), we must have that $t^* < r_1$. Otherwise, (t^*, j^*) could not have been picked as an initiator. We now complete the proof exactly along the lines of Corollary 4.1. □

COROLLARY 5.2. *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 $\overline{H}^i_{r_1,t}$ (i.e., $\widehat{H}^i_t \leq \overline{H}^i_{r_1,t}$).*

PROOF. Same as in Corollary 4.2. □

THEOREM 5.1. *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 $(\hat{x},\hat{y})$ toward 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 $\sum_{j\in \text{path}_{i,j^*}} \hat{z}^{ij}_{\mathsf{N}_{j^*}(r_1),t} > 0$, where $j^* \in \mathcal{S}(i)$ is the corresponding initiator item, and $\text{freeze}(i,t) \le \mathsf{N}_{j^*}(r_1) \le t$. In either case, this is clearly bounded by $\hat{b}^i_t$. If (i,t) contributes only toward a single order in R', then by Corollary 5.2, the holding cost it incurs is bounded by $\hat{b}^i_t$, and we can pay for its holding cost and its contributions using at most $2\hat{b}^i_t$.

Now assume that (i,t) also contributes toward 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 $r_2 \notin [\text{freeze}(i,t),t]$. If $j' \in \mathcal{S}(i)$ is the corresponding initiator item of this order, then (i,t) contributes $\sum_{j\in\text{path}_{ij'}} \hat{z}^{ij}_{\mathsf{N}_{j'}(r_2),t} > 0$ toward r_2, and $\text{freeze}(i,t) \le \mathsf{N}_{j'}(r_2) < r_1$ (see the proof of Corollary 5.1). We shall argue that

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

The first inequality follows from $\hat{z}^{ij}_{\mathsf{N}_{j'}(r_2),t} \ge 0$ $(\forall j \in \mathcal{S}(i))$, and the second inequality follows from the monotonicity of the holding costs and $\mathsf{N}_{j'}(r_2) < r_1$. From Corollary 5.2, we get that $\hat{b}^i_t \ge \hat{H}^i_t + \sum_{j\in\text{path}_{ij'}} \hat{z}^{ij}_{\mathsf{N}_{j'}(r_2),t}$. Corollary 5.1 and the fact that each demand point can contribute only toward orders $r \in R'$ with item i imply that (i,t) does not contribute toward 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 toward 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 nondecreasing function of the ordering time. Finally, as in the JRP, we have not been able to show whether the analysis is tight.

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 two-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 $(t = 1,\ldots,T)$, 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 two-stage assembly problem, and because we can restrict attention to nested policies, it is equivalent to the original JRP instance. Note that for this example, the conventional holding cost parameters are in fact identical to the echelon holding cost parameters (because there are only two stages).

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 LPs 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 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.

An 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 cannot even exclude the existence of a polynomial-time approximation scheme (i.e., one might be able to design a ρ-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. Two more specific open questions are related to the tightness of the analysis of the primal-dual algorithms and the LP relaxations considered in this paper. We have constructed (Roundy et al. [21]) an example in which the integrality gap is 1.23. This implies that using the LP as the only lower bound, one cannot hope to prove a performance guarantee better than 1.23. However, there still exists a significant gap between the upper bound of 2 and the lower bound of 1.23.

Acknowledgments. The authors thank Chaitanya Swamy for stimulating discussions and helpful suggestions and the anonymous referees for constructive comments. This research was conducted while Retsef Levi was a Ph.D. student in the School of Operations Research and Industrial Engineering at Cornell University and his research was partially supported by a grant from Motorola and NSF Grants CCR-9912422 and CCR-0430682. The research of Robin O. Roundy was partially supported by a grant from Motorola, NSF Grants DMI-0075627 and DMI-0500263, and the Querétaro Campus of the Instituto Tecnológico y de Estudios Superiores de Monterrey. The research of David B. Shmoys was partially supported by NSF Grants CCR-9912422, CCR-0430682, and DMI-0500263.

References

[1] Archer, A. 2000. Inapproximability of the asymmetric facility location and k-median problems. Working paper, Algorithm and Optimization Group, AT&T, Shannon Research Laboratory, Florham Park, NJ.
[2] Arkin, E., D. Joneja, R. Roundy. 1989. Computational complexity of uncapacitated multi-echelon production planning problems. *Oper. Res. Lett.* **8** 61–66.
[3] Askoy, Y., S. S. Erenguk. 1988. Multi-item inventory models with coordinated replenishment: A survey. *Internat. J. Oper. Production Management* **8** 63–73.
[4] Bárány, I., T. J. Van Roy, L. A. Wolsey. 1984. Uncapacitated lot-sizing: The convex hull of solutions. *Math. Programming Study* **22** 32–43.
[5] Bertsimas, D., C. Teo, R. Vohra. 1999. On dependent randomized rounding algorithms. *Oper. Res. Lett.* **25** 105–114.
[6] Bussieck, M. R., A. Fink, M. E. Lübbecke. 1998. 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, Braunschweig, Germany.
[7] Chuzhoy, Julia, Sudipto Guha, Eran Halperin, Sanjeev Khanna, Guy Kortsarz, Robert Krauthgamer, Joseph Naor. 2005. Asymmetric k-center is log$^*n$-hard to approximate. *J. ACM* **52**(4) 538–551.
[8] Crowston, W. B., M. H. Wagner. 1973. Dynamic lot size models for multi-stage assembly systems. *Management Sci.* **20** 14–21.
[9] Federgruen, A., M. Tzur. 1994. The joint replenishment problem with time-varying parameters: Efficient, asymptotic and epsilon-optimal solutions. *Oper. Res.* **42** 1067–1087.
[10] Goemans, M. X., D. P. Williamson. 1995. A general approximation technique for constrained forest problems. *SIAM J. Comput.* **24** 296–317.
[11] Jain, K., V. V. Vazirani. 2001. Approximation algorithms for metric facility location and k-median problems using the primal-dual schema and Lagrangian relaxation. *J. ACM* **48** 274–296.
[12] Joneja, D. 1987. Multi-echelon and joint replenishment production and distribution systems with nonstationary demand. Technical report 731, School of Operations Research and Industrial Engineering, Cornell University, Ithaca, NY.
[13] Joneja, D. 1989. Planning for joint replenishment and assembly systems with deterministic non-stationary demands. Ph.D. thesis, School of Operations Research and Industrial Engineering, Cornell University, Ithaca, NY.
[14] Joneja, D. 1990. The joint replenishment problem: New heuristics and worst case performance bounds. *Oper. Res.* **38** 723–771.
[15] Kao, E. P. C. 1979. A multi-product dynamic lot-size model with individual and joint set-up costs. *Oper. Res.* **27** 279–289.
[16] Krarup, J., O. Bilde. 1977. Plant location, set covering and economic lot sizing: An *O(mn)* algorithm for structural problems. L. Collatz, W. Wetterling, eds. *Numerische Methoden bei Optimierungsaufgaben—Band 3* (Optimierung bei Graphentheoretischen und Ganzzahligen Problemen), Vol. 36, *International Series of Numerical Mathematics.* Birkhauser Verlag, Basel, Switzerland, 155–180.
[17] Levi, R., R. O. Roundy. A note on Joneja's joint replenishment problem approximation algorithm. In preparation.
[18] Raghavan, P., M. R. Rao. 1991. The multi-item lot sizing problem with joint replenishment: A polyhedral approach. Technical report SOR-91-8, Stern School of Business, New York University, New York.
[19] Raghavan, P., M. R. Rao. 1992. Formulations to the multi-item lot sizing problem with joint replenishment. Technical report SOR-92-19, Stern School of Business, New York University, New York.
[20] Roundy, R. O. 1993. Efficient, effective lot-sizing for multi-product, multi-stage production systems. *Oper. Res.* **41** 371–386.
[21] Roundy, R. O., R. Levi, D. B. Shmoys. 2003. A lower bound on the integrality gap for a strong IP formulation for the joint replenishment problem. Working paper, Department of Operations Research and Industrial Engineering, Cornell University, Ithaca, NY.
[22] Shen, Z. J., D. Simchi-Levi, C. P. Teo. Approximation algorithms for the single-warehouse multi-retailer problem with piecewise linear cost structures. http://citeseer.nj.nec.com/439759.html.
[23] Shmoys, D. B., C. Swamy, R. Levi. 2004. Facility location with service installation costs. *Proc. 15th Annual SIAM-ACM Sympos. Discrete Algorithms*, New York, 1081–1090.
[24] van Hoesel, A., A. Wagelmans, A. Kolen. 1991. A dual algorithm for the economic lot-sizing problem. *Eur. J. Oper. Res.* **52** 315–325.
[25] Veinott, A. F. 1969. Minimum concave cost solutions of Leontief substitution models of multi-facility inventory systems. *Oper. Res.* **17** 262–291.
[26] Wagner, H. M., T. M. Whitin. 1958. Dynamic version of the economic lot sizing model. *Management Sci.* **5** 89–96.
[27] Zangwill, W. I. 1966. A deterministic multi-product, multi-facility production and inventory model. *Oper. Res.* **14** 486–507.