Approximation Algorithms for Inventory Problems with Submodular or Routing Costs

Viswanath Nagarajan*   Cong Shi*

Abstract

We consider the following two deterministic inventory optimization problems with non-stationary demands.

- *Submodular Joint Replenishment Problem.* This involves multiple item types and a single retailer who faces demands over a finite planning horizon of T periods. In each time period, any subset of item-types can be ordered incurring a joint ordering cost which is submodular. Moreover, items can be held in inventory while incurring a holding cost. The objective is to find a sequence of orders that satisfies all demands and minimizes the total ordering and holding costs.

- *Inventory Routing Problem.* This involves a single depot that stocks items, and multiple retailer locations facing demands over a finite planning horizon of T periods. In each time period, any subset of locations can be visited using a vehicle originating from the depot. There is also cost incurred for holding items at any retailer. The objective here is to satisfy all demands while minimizing the sum of routing and holding costs.

We present a unified approach that yields $\mathcal{O}\left(\frac{\log T}{\log \log T}\right)$-factor approximation algorithms for both problems when the holding costs are polynomial functions. A special case is the classic linear holding cost model, wherein this is the first sub-logarithmic approximation ratio for either problem.

Key words: Inventory Management, Approximation Algorithms, Submodular Function, Joint Replenishment Problem, Inventory Routing Problem

1   Introduction

Deterministic inventory theory provides streamlined optimization models that attempt to capture tradeoffs in managing the flow of goods through a supply chain. We consider two classical models in deterministic inventory theory: the *Joint Replenishment Problem* (JRP) and the *Inventory Routing Problem* (IRP). These inventory models have been studied extensively in the literature (see, e.g., [1], [21]) and recently there has been significant progress on many variants of these models (see, e.g., [23], [22], [24], [10], [17]). In this paper, we present a unified approach that yields approximation algorithms for both models with generalized cost structure – the JRP with submodular setup cost and the IRP with an arbitrary metric.

The JRP with deterministic and non-stationary demand is a fundamental yet notoriously difficult problem in inventory management. In these models, there are multiple item types, and we need to coordinate a sequence of (joint) orders to satisfy the demands for different item types before their respective due dates. Ordering inventory in a time period results in setup costs (or fixed ordering costs), and holding inventory before it is due results in holding costs. The objective is to find a feasible ordering policy to satisfy every demand point on time over a finite planning horizon so as to minimize the sum of setup and holding costs. The JRP is a natural extension of the classical economic lot-sizing model that considers the optimal trade-off between setup costs and holding costs for a single item type (see [32]). With multiple item types, the JRP adds the possibility of saving costs via coordinated replenishment, a common phenomenon in supply chain management.

---

*Industrial & Operations Engineering, University of Michigan, Ann Arbor, MI, {viswa, shicong}@umich.edu

Most of the literature on deterministic JRP is on the *additive* joint setup cost structure. Under this structure, there is a one-time setup cost if any item type is ordered, and there is an individual item setup cost for each item type ordered; the joint setup cost for this particular order is simply the sum of the one-time setup cost and these individual item setup costs. The additive joint setup cost structure loses significant amount of modeling power and flexibility (see [25], [13], [28], [10]). In this paper, we adopt the joint setup cost structure introduced recently in [10] that satisfies two natural properties known as *monotonicity* and *submodularity*. The monotonicity property means that the joint setup cost increases with the set of item types ordered. The submodularity property captures economies of scale in ordering more item types, i.e., the marginal cost of adding any specific item type to a given order decreases in the set of item types included.

The IRP is also a classical problem in inventory management that captures the trade-off between the holding costs for inventory and the routing costs for replenishing the inventory at various locations in a supply chain (see, e.g., [7], [12], [14], [17]). The problem involves multiple item types that are stocked in a single depot, that must be shipped to meet the demand for these item types arising at multiple retailers specified over the course of a planning horizon. Similar to the JRP, the costs of holding a unit of inventory at each retailer are specified to compute the inventory holding costs. Different than the JRP, we consider transportation (or vehicle routing) costs in some metric defined by the depot and retailers in the IRP, instead of joint setup costs considered in the JRP.

1.1 Main Results and Contributions

We present a unified approach that yields $\mathcal{O}\left(\frac{\log T}{\log\log T}\right)$-approximation algorithms for both the JRP with submodular setup costs and the IRP with an arbitrary metric (where T is the length of the planning horizon), when the holding costs are polynomial functions (which subsumes conventional linear costs as a special case). This is the first sub-logarithmic approximation ratio for either problem under these cost structures.

We remark that if the setup cost function in submodular-JRP is time-dependent then the problem (even with zero holding costs) becomes as hard to approximate as set cover [15]. The same observation is true if the metric in IRP is time-dependent. So our sub-logarithmic ratio approximation algorithm relies crucially on the uniformity of these costs over time.

For the submodular JRP, Cheung et al. [10] obtained constant-factor approximation algorithms under several special submodular functions (i.e., tree, laminar and cardinality). In contrast, we consider general submodular functions with special (polynomial) holding costs.

For the IRP, Fukunaga et al. [17] considered a restricted class of "periodic policies" and obtained a constant-factor approximation algorithm. Whereas our result is for arbitrary policies and polynomial holding costs.

A straightforward modification of our algorithm for polynomial holding costs also yields $\mathcal{O}(\log T)$-approximation algorithms for submodular JRP and IRP with arbitrary (monotone) holding costs. The submodular JRP result improves upon the approximation ratio of $\mathcal{O}(\log(NT))$ by [10]. The IRP result is incomparable to the $\mathcal{O}(\log n)$ approximation ratio mentioned in [17], where n is the number of retailers.

1.2 Our Approach

At a high-level, the algorithm for submodular JRP has the following steps. (The algorithm for IRP is very similar – we in fact present an algorithm for a unified problem formulation.) First, we solve a natural time-indexed LP relaxation that was also used in [10]. Then we construct a "shadow interval" for each demand point that corresponds to fractionally ordering half unit of the item. We also stretch each shadow interval appropriately (depending on the degree of the holding cost function) so as to obtain an optimal trade-off between holding and setup costs: this is what results in the $\mathcal{O}\left(\frac{\log T}{\log\log T}\right)$ approximation ratio. Next, we partition these stretched intervals into multiple groups based on well-separated widths. Finally we place a separate sequence of orders for each group, and argue

using submodularity of the setup cost function that the total setup cost of each group is bounded by the LP setup cost. This step relies on the *fractional subadditivity* property of submodular functions.

It turns out that we do not require the full strength of submodular functions: the algorithm and analysis work even for functions satisfying an approximate notion of fractional subadditivity (see Definition 2.1) as long as the natural LP relaxation can be solved approximately. This allows us to also obtain an approximation algorithm for IRP since the TSP cost function satisfies 1.5-approximate fractional subadditivity and there is a $2 + o(1)$ approximation algorithm for its LP relaxation (see Section 4 for details).

We believe that some of our techniques may be useful in obtaining a constant factor approximation algorithm for both problems in their full generality.

1.3 Literature review

As mentioned earlier, most of the existing literature on deterministic JRP with non-stationary demand uses the additive joint setup cost structure. Arkin et al. [4] showed that the additive JRP is NP-hard. Nonner and Sviridenko [24] further showed that the additive JRP is in fact APX-hard with nonlinear holding cost structure. There have been several approximation algorithms for the additive JRP (see [10] and the references therein). The state-of-the-art approximation algorithms for the additive JRP are due to [23], [22] and [6], with approximation ratios 2, 1.80 and 1.791, respectively.

Due to the limited modeling power of the additive JRP, Cheung et al. [10] first studied the submodular JRP in which the setup costs are submodular. They gave an $\mathcal{O}(\log(NT))$-approximation algorithm for the general submodular JRP (where N is number of items and T is number of periods). They also analyzed three special cases of submodular functions which are laminar, tree and cardinality costs. They showed that the laminar case can be solved optimally in polynomial time using dynamic programming, and obtained a 3-approximation for the tree case and a 5-approximation for the cardinality case. Our work contributes to the literature by giving approximation algorithms for the general submodular JRP with special holding cost structures.

The IRP has also been studied extensively in the literature (see [7], [12], [14], [17] for an overview of this problem). The problem can be cast as a mathematical program (see, e.g., [8]) and solution approaches typically involve heuristics that trade-off between holding and transportation costs (see [2, 3], [9], [11], [31], [5]). Closer to our work, Fukunaga et al. [17] gave constant factor approximation algorithms for the IRP restricting to periodic schedules. In contrast, our results do not require the schedule to be periodic but require polynomial holding costs.

1.4 Structure of this paper and some notations

We organize the remainder of the paper as follows. In Section 2, we present a unified formulation for the submodular JRP and the IRP with an arbitrary metric, and state our main result. In Section 3, we propose a unified approximation algorithm for both problems. In Section 4, we discuss how to solve the LP relaxation efficiently. We conclude our paper in Section 5.

Throughout the paper, we use the notation $\lfloor x \rfloor$ and $\lceil x \rceil$ frequently, where $\lfloor x \rfloor$ is defined as the largest integer value which is smaller than or equal to x; and $\lceil x \rceil$ is defined as the smallest integer value which is greater than or equal to x. Additionally, for any real numbers x and y, we denote $x^+ = \max\{x, 0\}$, $x \vee y = \max\{x, y\}$, and $x \wedge y = \min\{x, y\}$. The notation := reads "is defined as".

2 A Unified Formulation for the JRP and the IRP

In this section, we formally describe a unified problem statement that includes two classical deterministic inventory problems as special cases, i.e., the submodular joint replenishment problem (JRP) and the inventory routing

problem (IRP). We also present an integer programming formulation for the unified problem, and state our main result.

2.1 Problem Statement

There are N elements (e.g., item types in the JRP or retailers in the IRP) that are needed to serve external demands over a finite planning horizon of T periods; these elements are denoted by the ground set $\mathcal{N} = \{1, \dots, N\}$, and the time periods are denoted by the set $\mathcal{T} = \{1, \dots, T\}$. For each time period $t \in \mathcal{T}$ and each element $i \in \mathcal{N}$, there is a known demand $d_{it} \geq 0$ units of that element. We use $\mathcal{D}$ to denote the set of all strictly positive demand points (i, t) with $d_{it} > 0$. To satisfy these demands, an order may be placed in each time period. Each demand point $(i, t) \in \mathcal{D}$ has to be served by an order containing element i before or at time period t, i.e., no backlogging or lost-sales are allowed.

The inventory system incurs two types of cost – the joint ordering cost and the holding cost.

- The joint ordering cost is a function of the elements that place strictly positive orders in any given period. More specifically, for any time period t and a subset of elements $S \subseteq \mathcal{N}$, the joint ordering cost of ordering demand for elements in S in period t is a function of S, which is denoted by $f(S)$. We assume that f is monotone (i.e. $f(A) \leq f(A')$ for all $A \subseteq A' \subseteq \mathcal{N}$) and subadditive (i.e. $f(A \cup B) \leq f(A) + f(B)$ for all $A, B \subseteq \mathcal{N}$). This is satisfied in most applications.

- Because the setup cost of ordering an element 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 by holding inventory over time periods. We use h^i_{st} to denote the holding cost incurred by ordering one unit of inventory in period s, and using it to meet the demand for element i in period t. We assume that h^i_{st} is non-negative and, for each demand point (i, t), is a nonincreasing function of s, i.e., holding inventory longer is always more costly. Thus, if the demand point (i, t) is served by an order at time period s, then the system incurs a holding cost of $H^i_{st} := d_{it} h^i_{st}$.

The goal is to coordinate a sequence of (joint) orders to satisfy all the demand points on time so as to minimize the sum of joint ordering and holding costs over the T periods.

The above unified problem statement encompasses two classical deterministic inventory problems described below.

Submodular joint replenishment problem. The JRP involves multiple item types and a single retailer who faces demands. In each time step, any subset of item-types can be ordered incurring a joint ordering cost which is submodular. The objective is to find a sequence of orders that satisfies all demands and minimizes the total ordering and holding costs. The elements in the above problem statement are the item types in the JRP; and the joint ordering cost $f(\cdot)$ is commonly referred to as the setup cost (or equivalently, the fixed ordering cost) in the JRP.

The submodular JRP considers a special class of $f(\cdot)$ called submodular functions (see, e.g., [10]). More precisely, we assume that the function $f(\cdot)$ is non-negative, monotone non-decreasing, and also submodular. The non-negativity and monotonicity assert that for every $S_1 \subseteq S_2 \subseteq \mathcal{N}$, we have $0 \leq f(S_1) \leq f(S_2)$. Submodularity requires that for every set $S_1, S_2 \subseteq \mathcal{N}$, we have

$$f(S_1) + f(S_2) \geq f(S_1 \cup S_2) + f(S_1 \cap S_2).$$

There is an equivalent definition that conveys the economies of scale more clearly. That is, for every set $S_1 \subseteq S_2 \subseteq \mathcal{N}$ and any item type $i \in \mathcal{N}$, we have $f(S_2 \cup \{i\}) - f(S_2) \leq f(S_1 \cup \{i\}) - f(S_1)$, i.e., the additional cost of adding an item type to the joint order is decreasing as more item types have been included in that order.

Inventory routing problem. The IRP involves a single depot r that stocks items, and a set of retailer locations (denoted by the ground set $\mathcal{N}$) facing demands. In each time step, any subset of locations can be visited using

a vehicle originating from the depot. The objective here is to satisfy all demands while minimizing the sum of routing and holding costs. The elements in the above unified problem statement are the retailers in the IRP; and the joint ordering cost $f(\cdot)$ is the shipping or routing cost.

The IRP is specified by a complete graph on vertices V with a metric distance function $w : \binom{V}{2} \to \mathbb{R}_+$ that satisfies symmetry (i.e. $w(ba) = w(ab)$ for any $a, b \in V$) and triangle inequality (i.e. $w(ab) + w(bc) \geq w(ac)$ for any $a, b, c \in V$). The vertex set $V = \mathcal{N} \cup r$, containing the depot and the set of retailers. The shipping or routing cost $f(S)$ is defined as the travelling salesman (TSP) cost of visiting the retailers in $S \subseteq V$. Formally,

$$f(S) \quad := \quad \text{minimum length of tour that visits each vertex in } S \cup \{r\}, \qquad \forall S \subseteq \mathcal{N}. \tag{1}$$

2.2 IP Formulation and its LP Relaxation

The unified problem described above can be written as an integer programming problem as follows (see also [10]). First we define two types of binary variables y_s^S and x_{st}^i such that

$$y_s^S \quad = \quad \begin{cases} 1, & \text{if the subset of elements } S \subseteq \mathcal{N} \text{ is ordered in period } s, \\ 0, & \text{otherwise.} \end{cases}$$

$$x_{st}^i \quad = \quad \begin{cases} 1, & \text{if the demand point } (i, t) \text{ is satisfied using an order from period } s, \\ 0, & \text{otherwise.} \end{cases}$$

Then the integer programming (IP) formulation is given by

$$\textbf{(IP)} \qquad \min \qquad \sum_{S \subseteq \mathcal{N}} \sum_{s=1}^{T} f(S) y_s^S + \sum_{(i,t) \in \mathcal{D}} \sum_{s=1}^{t} H_{st}^i x_{st}^i \tag{2}$$

$$\text{s.t.} \qquad \sum_{s=1}^{t} x_{st}^i \quad = \quad 1, \qquad \forall (i, t) \in \mathcal{D} \tag{3}$$

$$x_{st}^i \quad \leq \quad \sum_{S : i \in S \subseteq \mathcal{N}} y_s^S, \qquad \forall (i, t) \in \mathcal{D}, \forall s = 1, \ldots, t \tag{4}$$

$$x_{st}^i, y_s^S \quad \in \quad \{0, 1\}, \qquad \forall (i, t) \in \mathcal{D}, \forall s = 1, \ldots, t, \forall S \subseteq \mathcal{N}, \tag{5}$$

where (3) enforces that every demand point (i, t) must be served by an order before or in time period t, and (4) ensures that the joint order S must contain element i if any demand (i, t) is served from time period s. The natural linear programming (LP) relaxation of (IP) relaxes the integer constraints on x_{st}^i and y_s^S to non-negativity constraints.

2.3 Definitions and Assumptions

To obtain approximation algorithms for the IP (2) using our framework, we only need to assume that the set function $f(\cdot)$ satisfies an approximate notion of fractional subadditivity (which is much weaker than submodularity).

Definition 2.1 (β-approximate fractional subadditivity) *The set function $f(\cdot)$ is β-approximate fractional subadditive, if for any collection $\{S_i, \lambda_i\}$ of weighted subsets with $0 \leq \lambda_i \leq 1$ and $\sum_{i|v \in S_i} \lambda_i \geq 1$ for each $v \in S$, we have $f(S) \leq \beta \cdot \sum \lambda_i f(S_i)$. Namely, if the sets S_i form a fractional cover of $S \subseteq \mathcal{N}$, then the cost of S is at most β times the sum of the costs $f(S_i)$ weighted by the corresponding coefficients.*

It is known that if a function is submodular, then it is also fractional subadditive (see [16]), i.e., the notion of submodularity is stronger. For the submodular JRP, the setup cost function $f(S)$ is submodular and hence also fractional subadditive (or equivalently, 1-approximate fractional subadditive).

5

For the IRP with an arbitrary metric, the vehicle routing cost $f(S)$ although not submodular, can be shown to be 1.5-approximate fractional subadditive. This follows from the fact that the natural LP-relaxation for TSP has an integrality gap at most 1.5 (see [33] and [27]).

Note that the LP relaxation of (2) has an exponential number of variables; we need to ensure that this LP relaxation can be (at least approximately) solved efficiently.

Definition 2.2 (γ-approximate LP solution) *We say that a feasible LP solution is γ-approximate, if its objective value is at most γ times the optimal LP objective value.*

Using the ellipsoid method one can compute efficiently: an exact LP solution for the submodular JRP and a $(2 + o(1))$-approximate LP solution for IRP. We delegate the discussion on this to Section 4 for better readability of this paper.

Our results require that all holding costs be polynomial functions. Recall that for any integer $\alpha \geq 1$, an α-*degree polynomial* is a function $p(x) = \sum_{a=0}^{\alpha} c_a \cdot x^a$ where each coefficient $c_0, \cdots, c_\alpha \in \mathbb{R}$.

Definition 2.3 (α-degree polynomial holding cost) *The holding cost $h_{\star t}^i$ of any demand $(i, t) \in \mathcal{D}$ is said to be an α-degree polynomial if $h_{st}^i = p_{it}(t - s)$ for all $s \leq t$, where $p_{it}(\cdot)$ is an α-degree polynomial with non-negative coefficients.*

When $\alpha = 1$, this reduces to the conventional linear holding cost.

2.4 Main Results

Now we are in a position to formally state our main result (which will be proved in the following section).

Theorem 2.4 *There is an $\mathcal{O}\left(\alpha\beta\gamma \cdot \frac{\log T}{\log \log T}\right)$-approximation algorithm for the integer program defined in (2), provided that (i) $f(\cdot)$ is β-approximate fractional subadditive, (ii) all holding costs are α-degree polynomials, and (iii) a γ-approximate solution to the LP relaxation of (2) can be found in polynomial time.*

Corollary 2.5 below is an immediate consequence of Theorem 2.4, since

1. $\beta = \gamma = 1$ for the submodular JRP;

2. $\beta = 1.5$ and $\gamma = 2 + o(1)$ for the IRP.

Corollary 2.5 *There is an $\mathcal{O}\left(\frac{\log T}{\log \log T}\right)$-approximation algorithm for both the submodular JRP and the IRP when all holding costs are α-degree polynomials.*

To the best of our knowledge, this is the first sub-logarithmic approximation ratio for either problem.

We remark that if one considers arbitrary (monotone) holding costs (instead of just polynomials), our approach yields $\mathcal{O}(\log T)$-approximation algorithms for the submodular JRP and the IRP (see Section 3.3).

3 LP-Rounding Algorithm

We present an LP-rounding algorithm for the integer program (2) in Section 3.1, and then carry out a worst-case performance analysis in Section 3.2.

3.1 Algorithm Description

We describe our procedure of rounding a γ-approximate solution (y, x) of (LP). We set $\rho := \lfloor (\log T)^{1/(2\alpha)} \rfloor$.

Step 1 – Constructing extended shadow intervals.   We first construct what-we-call *extended shadow intervals* as follows. For each demand point (i, t), we find a time period $s'(i, t)$ such that

$$\sum_{s=s'(i,t)}^{t} x_{st}^i \geq 1/2 \text{ and } \sum_{s=s'(i,t)+1}^{t} x_{st}^i < 1/2,$$

Then $[s'(i, t), t]$ is called the *shadow interval* for this particular demand point (i, t). We also measure its length $l_{(i,t)} := t - s'(i, t)$.

Next for each demand point (i, t), we round the length $l_{(i,t)}$ up to the nearest power of ρ. If $s'(i, t) = t$ then we set $s^*(i, t) = t$. Else we find the smallest integer $m \geq 1$ such that $\rho^m \geq t - s'(i, t)$, and then stretch the shadow interval until $s^*(i, t)$ where

$$s^*(i, t) := t - \rho^m \leq s'(i, t).$$

We call the interval $[s^*(i, t), t]$ the *extended shadow interval* for the demand point (i, t), and also measure its length $l^*(i, t) := t - s^*(i, t)$. Figure 1 below gives a graphical representation of this step.
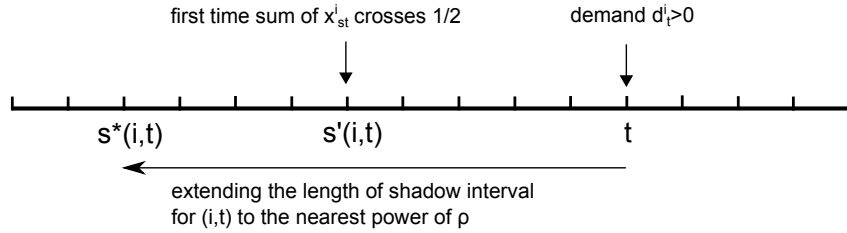

first time sum of x_{st}^i crosses 1/2   demand $d_t^i > 0$

s*(i,t)   s'(i,t)   t

extending the length of shadow interval
for (i,t) to the nearest power of ρ

Figure 1: Illustration of an extended shadow interval for demand point (i, t).

Step 2 – Partitioning demand points according to extended shadow intervals.   Next we partition the demand points according to the length of their extended shadow intervals. For each demand point (i, t), its length $l^*(i, t)$ falls into exactly one of the values below (recall by construction $l^*(i, t)$ is either zero or an integer power of ρ).

$$\{0, \rho^1, \rho^2, \ldots, \rho^{k-2}, \rho^{k-1} \wedge T\}, \text{ where } k = 1 + \lceil \log_\rho T \rceil.$$

In this way, we have partitioned the demand points into $k = \mathcal{O}\left(\alpha \frac{\log T}{\log \log T}\right)$ number of groups as follows,

$$\mathcal{L}_0 = \{(i, t) \in \mathcal{D} : l^*(i, t) = 0\} \quad \text{and} \quad \mathcal{L}_m = \{(i, t) \in \mathcal{D} : l^*(i, t) = \rho^m \wedge T)\}, \ \forall \, m \in \{1, \ldots, k-1\}.$$

That is, the extended shadow intervals within each group $\mathcal{L}_m$ share the same length:

$$w_m := \begin{cases} 0 & \text{if } m = 0 \\ \rho^m \wedge T & \text{if } m \in \{1, 2, \ldots, k-1\} \end{cases}$$

Step 3 – Placing orders.   Based on the above partition of demand points, we describe our ordering procedure.

Now fix an $m \in \{0, 1, \ldots, k-1\}$ and focus on the demand group $\mathcal{L}_m$. Let $\tau_j = 1 + j \cdot w_m \ (\leq T)$ for $j = 0, 1, \ldots$. We place a tentative (joint) order in each period τ_j $(j = 0, 1, \ldots)$, i.e., once every w_m periods. In each period $\tau_j \leq T$ $(j = 0, 1, 2 \ldots)$, we identify the set of elements

$$A_m^j = \{i : (i, t) \in \mathcal{L}_m \text{ and } \tau_j \in [s^*(i, t), t]\},$$

7

i.e., all the elements within $\mathcal{L}_m$ whose extended shadow interval contains time period τ_j. We then place an actual joint order that serves the demand points associated with A_m^j in period τ_j. Figure 2 gives one specific example of how the algorithm places these orders.

We repeat the above procedure for all groups $m = 0, 1, \ldots, k-1$. In any given period, if there is more than one joint order (across different groups), we simply merge them into a single joint order.
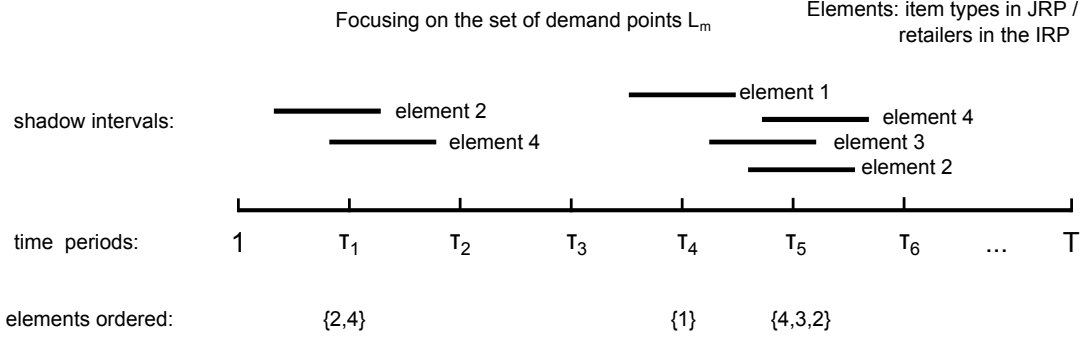
Focusing on the set of demand points L$_m$ — Elements: item types in JRP / retailers in the IRP

Figure 2: Placing actual orders for the demand points within $\mathcal{L}_m$

This concludes the description of our LP-rounding algorithm.

3.2 Worst-case Analysis

We shall prove that our LP-rounding algorithm gives an $\mathcal{O}\left(\alpha\beta\gamma \frac{\log T}{\log\log T}\right)$-approximation for the unified problem.

Analysis of Holding Cost

Lemma 3.1 *Let $[s'(i,t), t]$ be the shadow interval for some demand point (i,t). Then we have*

$$H^i_{s'(i,t)t} \leq 2 \cdot \sum_{s=1}^{t} H^i_{st} x^i_{st}, \tag{6}$$

where x^i_{st}'s are in the γ-approximate LP solution.

Proof: Since we deal with a fixed demand (i,t) here, we use $s' = s'(i,t)$ to reduce notation. By the construction of shadow intervals, we have $\sum_{s=s'+1}^{t} x^i_{st} < 1/2$. Now since $\sum_{s=1}^{t} x^i_{st} = 1$ by (3), we must have $\sum_{s=1}^{s'} x^i_{st} \geq 1/2$. Hence we have

$$H^i_{s't} \leq 2 \cdot \sum_{s=1}^{s'} H^i_{s't} x^i_{st} \leq 2 \cdot \sum_{s=1}^{s'} H^i_{st} x^i_{st} \leq 2 \cdot \left(\sum_{s=1}^{s'} H^i_{st} x^i_{st} + \sum_{s=s'+1}^{t} H^i_{st} x^i_{st} \right) = 2 \cdot \sum_{s=1}^{t} H^i_{st} x^i_{st}, \tag{7}$$

where the second inequality is due to $H_{s't} \leq H_{st}$ for all $s \leq s'$ by monotonicity of holding costs. ∎

Lemma 3.2 *The total holding cost for the solution found by the LP-rounding algorithm is at most*

$$\mathcal{O}(\sqrt{\log T}) \cdot \sum_{(i,t)\in\mathcal{D}} \sum_{s=1}^{t} H^i_{st} x^i_{st},$$

where x^i_{st}'s are in the γ-approximate LP solution.

8

Proof: Fix any $m \in \{0, 1, \cdots, k-1\}$ and demand point (i, t) in $\mathcal{L}_m$. To reduce notation we use $s' = s'(i, t)$ and $s^* = s^*(i, t)$ below. By the polynomial holding cost structure, we have

$$
\begin{aligned}
H_{s^*t}^i &= d_t^i \left(c_\alpha w_m^\alpha + c_{\alpha-1} w_m^{\alpha-1} + \ldots + c_1 w_m + c_0 \right), \\
H_{s't}^i &\geq d_t^i \left(c_\alpha (w_m/\rho)^\alpha + c_{\alpha-1}(w_m/\rho)^{\alpha-1} + \ldots + c_1(w_m/\rho) + c_0 \right),
\end{aligned}
$$

where all coefficients $c_0, \cdots, c_\alpha \geq 0$. The inequality above follows from the construction of extended shadow intervals. Hence it is clear that

$$
H_{s^*t}^i \leq \rho^\alpha \cdot H_{s't}^i.
$$

By the LP-rounding algorithm, for each demand point (i, t), we must have placed a (joint) order containing element i inside its extended shadow interval $[s^*, t]$. Due to monotonicity of holding costs, the worst-case (that gives the highest possible holding cost) happens when our algorithm places the order at exactly time period s^* to satisfy the demand point (i, t). Hence, the total holding cost associated with the demand point (i, t) is upper bounded by

$$
H_{s^*t}^i \leq \rho^\alpha H_{s't}^i \leq 2\rho^\alpha \cdot \sum_{s=1}^{t} H_{st}^i x_{st}^i,
$$

where the second inequality follows from Lemma 3.1. Now setting $\rho = \lfloor (\log T)^{1/(2\alpha)} \rfloor$ and adding over all demand points $(i, t) \in \mathcal{D}$ yields the result. $\blacksquare$

The intuition behind Lemma 3.2 is that when we stretch the shadow interval to the nearest integer power of ρ, the holding cost within the extended shadow interval does not grow too large due to the polynomial holding cost structure. On the other hand, stretching the shadow intervals in this manner allows for a tighter bound on the ordering cost (as shown next).

Analysis of Ordering Cost

To analyze the ordering cost component, we introduce the following bridging problem:

$$
\textbf{(Covering-LP)} \quad \min \quad \sum_{S \subseteq \mathcal{N}} \sum_{s=1}^{T} f(S) z_s^S \tag{8}
$$

$$
\text{s.t.} \quad \sum_{s=s^*(i,t)}^{t} \sum_{S: i \in S \subseteq \mathcal{N}} z_s^S \geq 1, \qquad \forall (i,t) \in \mathcal{D}
$$

$$
z_s^S \geq 0, \qquad \forall s = 1, \ldots, t, \forall S \subseteq \mathcal{N}.
$$

The intuition behind introducing this bridging problem is as follows: if our algorithm places an order to satisfy each demand anywhere within its extended shadow interval, then Lemma 3.2 already implies that the holding cost can be bounded by $\mathcal{O}(\sqrt{\log T})$ times the LP holding cost. Thus, it only remains to bound the ordering cost, which reduces to finding a "cover" for these extended shadow intervals as defined in Problem (8). We now focus on analyzing this Covering-LP.

Lemma 3.3 *The optimal objective value of the Covering-LP is at most*

$$
2 \cdot \sum_{S \subseteq \mathcal{N}} \sum_{s=1}^{T} f(S) y_s^S.
$$

where y_s^S's are in the γ-approximate LP solution.

9

Proof: We first check $\bar{z}_s^S = 2y_s^S$ (where y_s^S is the γ-approximate LP solution) is feasible to the Covering-LP defined in (8). It is obvious that $\bar{z}_s^S = 2y_s^S \geq 0$ for all $s = 1, \ldots, t$ and for all $S \subseteq \mathcal{N}$. It suffices to verify the first set of constraints. Indeed, for each $(i, t) \in \mathcal{D}$, we have

$$\sum_{s=s^*(i,t)}^{t} \sum_{S:i\in S\subseteq\mathcal{N}} \bar{z}_s^S = \sum_{s=s^*(i,t)}^{t} \sum_{S:i\in S\subseteq\mathcal{N}} 2y_s^S \geq \sum_{s=s^*(i,t)}^{t} 2x_{st}^i \geq 1, \tag{9}$$

where the first inequality follows from (4), and the second inequality follows from the fact that $\sum_{s=s^*(i,t)}^{t} x_{st}^i \geq \sum_{s=s'(i,t)}^{t} x_{st}^i \geq 1/2$ (by the definition of shadow intervals and their extensions).

Hence, the optimal objective value of the Covering-LP

$$\sum_{S\subseteq\mathcal{N}} \sum_{s=1}^{T} f(S)z_s^S \leq \sum_{S\subseteq\mathcal{N}} \sum_{s=1}^{T} f(S)\bar{z}_s^S = 2 \sum_{S\subseteq\mathcal{N}} \sum_{s=1}^{T} f(S)y_s^S, \tag{10}$$

where the first inequality follows from that z_s^S is optimal while $\bar{z}_s^S$ is feasible. ∎

Fix an $m \in \{0, 1, \ldots, k-1\}$ and we focus on the demand group $\mathcal{L}_m$ (which have equal length of extended shadow intervals). We shall show that the total ordering cost associated with the set $\mathcal{L}_m$ by our LP-rounding algorithm can be upper bounded by 2β times the Covering-LP cost. Our proof strategy relies on the notion of approximate fractional subadditivity (see Definition 2.1).

Lemma 3.4 *The total ordering cost associated with the set $\mathcal{L}_m$ by our LP-rounding algorithm is at most*

$$2\beta \cdot \sum_{S\subseteq\mathcal{N}} \sum_{s=1}^{T} f(S)z_s^S.$$

where z_s^S's are the optimal Covering-LP solutions.

Proof: Recall that for each demand group $\mathcal{L}_m$, the LP-rounding algorithm places a tentative (joint) order in each period $\tau_j \leq T$ ($j = 0, 1, \ldots$). Then in each time period τ_j the algorithm identifies the elements A_m^j within $\mathcal{L}_m$ whose extended shadow interval contains τ_j and places an actual (joint) order A_m^j that includes all of these "intersecting" elements.

Now take any $\tau_j \leq T$: since the length of the extended shadow intervals in $\mathcal{L}_m$ is exactly w_m, all the extended shadow intervals associated with the order A_m^j must lie within the interval $(\tau_{j-1}, \tau_{j+1}]$ (see Figure 2 as an example). Our LP-rounding algorithm places an actual (joint) order A_m^j in period τ_j and incurs an ordering cost $f(A_m^j)$. We will show that the Covering-LP provides us with a fractional cover of A_m^j which will be used to bound $f(A_m^j)$.

Indeed, for each demand point (i, t) associated with the order A_m^j, we have

$$\sum_{S:i\in S\subseteq\mathcal{N}} \sum_{s>\tau_{j-1}}^{\tau_{j+1}} z_s^S = \sum_{s>\tau_{j-1}}^{\tau_{j+1}} \sum_{S:i\in S\subseteq\mathcal{N}} z_s^S \geq \sum_{s=s^*(i,t)}^{t} \sum_{S:i\in S\subseteq\mathcal{N}} z_s^S \geq 1, \tag{11}$$

where the first inequality holds because every extended shadow interval associated with the order A_m^j must lie within the interval $(\tau_{j-1}, \tau_{j+1}]$ and the last inequality follows from the first constraint in the Covering-LP (8).

Since $f(\cdot)$ is β-approximate fractional subadditive, then according to Definition 2.1,

$$f(A_m^j) \leq \beta \cdot \sum_{S:S\subseteq\mathcal{N}} \sum_{s>\tau_{j-1}}^{\tau_{j+1}} z_s^S f(S). \tag{12}$$

10

It is then immediate that the ordering cost associated with the set $\mathcal{L}_m$ by our LP-rounding algorithm

$$\sum_{j \geq 0} f(A_m^j) \; \leq \; \beta \cdot \sum_{S:S\subseteq\mathcal{N}} \sum_{j \geq 0} \sum_{s > \tau_{j-1}}^{\tau_{j+1}} z_s^S f(S) \; \leq \; 2\beta \cdot \sum_{S:S\subseteq\mathcal{N}} \sum_{s=1}^{T} z_s^S f(S). \tag{13}$$

$\blacksquare$

Lemma 3.5 *The total ordering cost for the solution by our LP-rounding algorithm is at most*

$$\mathcal{O}\left(\alpha\beta \frac{\log T}{\log\log T}\right) \cdot \sum_{S\subseteq\mathcal{N}} \sum_{s=1}^{T} f(S) y_s^S,$$

where y_s^S's are in the γ-approximate LP solution.

Proof: Note that if in any time period there are orders from multiple groups $\{\mathcal{L}_m\}_{m=0}^{k-1}$ then the combined ordering cost is at most the sum of the ordering costs for each group: this uses the subadditivity property of $f(\cdot)$, i.e. $f(A \cup B) \leq f(A) + f(B)$ for any $A, B \subseteq \mathcal{N}$.

By Lemmas 3.3 and 3.4, for each group $\mathcal{L}_m$ $(m = 0, 1, \ldots, k-1)$, we conclude that the total ordering cost associated with the set $\mathcal{L}_m$ in our LP-rounding algorithm is at most

$$2\beta \cdot \sum_{S\subseteq\mathcal{N}} \sum_{s=1}^{T} f(S) z_s^S \leq 4\beta \cdot \sum_{S\subseteq\mathcal{N}} \sum_{s=1}^{T} f(S) y_s^S,$$

where z_s^S's are the optimal Covering-LP solution and y_s^S's are the γ-approximate LP solution. Then the result follows from the fact that the number of groups $k = \mathcal{O}\left(\alpha \frac{\log T}{\log\log T}\right)$. $\blacksquare$

Now we are ready to complete the proof of our main result.

Proof of Theorem 2.4: Combining the results from Lemmas 3.2 and 3.5, the total holding and ordering costs for the solution by our LP-rounding algorithm is at most $\mathcal{O}\left(\alpha\beta \frac{\log T}{\log\log T}\right)$ times the γ-approximate LP solution. $\blacksquare$

Remark: The $\mathcal{O}\left(\frac{\log T}{\log\log T}\right)$ approximation ratio is the best tradeoff achievable (in our approach) between the loss in holding and ordering costs, even under linear holding costs. Recall that for a given set W of widths for extended shadow intervals, the loss in ordering cost is just the number $|W|$ of distinct widths and the loss in holding cost depends on the aggregate stretch-factor incurred when the width of each shadow interval is increased to a value in W. Even if we allow for an arbitrary set W of widths (that may depend on the LP solution) and compute the worst ratio (using a "factor revealing linear program" as in [19]) then we obtain $\mathcal{O}\left(\frac{\log T}{\log\log T}\right)$ as the approximation ratio.

3.3 Immediate Extensions

General Holding Costs. If were to use our algorithm for arbitrary (monotone) holding costs, then we construct the shadow intervals in Step 1 but do not stretch them (i.e. we skip Step 2), and then we partition the shadow intervals using powers of *two*. A simplified version of our analysis then yields the following result.

Proposition 3.6 *There is an $\mathcal{O}(\log T)$-approximation algorithm for the submodular JRP and the IRP under monotone holding costs.*

A Special Case with Perishable Goods. We now consider a special holding cost which models perishable items with a fixed life-time $c > 0$. For each demand point (i, t), we can only start satisfying this order c periods before t, i.e., the ordering window is $[t - c, t]$. This setting is equivalent to the following holding cost structure. For each $i \in \mathcal{N}$ and $1 \leq s \leq t \in \mathcal{T}$,

$$h_{st}^i = \begin{cases} 0 & \text{if } t - c \leq s \leq t, \\ \infty & \text{if } s < t - c. \end{cases}$$

We also have $H_{st}^i = d_t^i h_{st}^i$.

In this setting, for each demand point (i, t), the extended shadow interval is simply $[t - c, t]$ with length c. Hence our LP-rounding algorithm and its worst-analysis will apply with just a single group, and we obtain:

Proposition 3.7 *When items are perishable with a fixed life-time and the holding cost is negligible, our algorithm gives a 2-approximation for the submodular JRP, and a $(6 + o(1))$-approximation for the IRP.*

Note that the assumption of uniform lifetime c across different items is necessary for the above result to hold.

4 Solving the LP Relaxation

As mentioned earlier in Section 2, the LP relaxation has an exponential number of variables and we first argue that there is an efficient way of solving this LP. We can readily write the dual of (LP) as

$$\textbf{(DLP)} \qquad \max \qquad \sum_{(i,t) \in \mathcal{D}} b_t^i \qquad\qquad\qquad (14)$$

$$\text{s.t.} \quad b_t^i \leq H_{st}^i + \bar{b}_{st}^i, \qquad \forall (i, t) \in \mathcal{D}, \forall s = 1, \ldots, t$$

$$f(S) - \sum_{i \in S} \sum_{t=s}^T \bar{b}_{st}^i \geq 0, \qquad \forall s = 1, \ldots, T, \forall S \subseteq \{1, \ldots N\}$$

$$\bar{b}_{st}^i \geq 0, \qquad \forall (i, t) \in \mathcal{D}, \forall s = 1, \ldots, t.$$

The b_t^i and $\bar{b}_{st}^i$ are the dual variable corresponding to the first and second constraints in the LP relaxation of (2). Note that the dual formulation (14) has an exponential number of constraints.

Submodular JRP. In the submodular JRP, the left hand side of the second constraint for any fixed s, $f(S) - \sum_{i \in S} \sum_{t=s}^T \bar{b}_{st}^i$ is clearly a submodular function on $\mathcal{N}$. Thus, there is an efficient separation oracle by using submodular function minimization [26] to find violated constraints. This implies that the dual problem (and therefore the primal) can be efficiently solved using the ellipsoid method. This was also discussed in [10].

Approximately Solving the LP for IRP. The TSP costs $f(\cdot)$ are not submodular, and in fact the above separation problem is NP-hard. However, there is an approximate separation oracle (see Lemma 4.3) which suffices to compute an approximately optimal solution to (DLP). The main ingredient is an approximation algorithm for the following auxiliary problem:

Definition 4.1 (Minimum ratio TSP) *The input is a metric (V, w) with a designated depot $r \in V$ and rewards $a : V \to \mathbb{R}_+$ on vertices. The goal is to find a subset $S \subseteq V$ that minimizes:*

$$\frac{f(S)}{a(S)}, \qquad \text{where } f(S) \text{ is the TSP cost as defined in (1) and } a(S) = \sum_{i \in S} a_i.$$

Theorem 4.2 (Garg [18]) *There is a $(2 + o(1))$-approximation algorithm for the minimum ratio TSP problem.*

Proof: The algorithm for minimum ratio TSP uses the 2-approximation algorithm for the related k-TSP problem: given a metric, depot r and target k, find a minimum length tour from the depot that visits at least k vertices. The algorithm which is based on standard scaling arguments, is given below for completeness:

1. Guess (by enumerating over $|V|$ choices) the maximum reward vertex u in an optimal solution.

2. Remove all vertices with reward more than a_u.

3. For each $v \in V$ set its new reward $\bar{a}_v$ to be the largest integer such that $\bar{a}_v \cdot a_u / n^2 \leq a_v$.

4. For each $k = 1, \cdots, n^3$, run the k-TSP algorithm with target k on the modified metric containing $\bar{a}_v$ co-located vertices at each $v \in V$.

5. Output the best ratio solution found (over all choices of u and k).

It is easy to see that this algorithm runs in polynomial time since each $\bar{a}_v \leq n^2$. We now show that it has an approximation ratio of $\gamma = 2 + o(1)$. Let S^* denote an optimal solution and $u \in S^*$ the maximum reward vertex. Consider the run of the above algorithm for this choice of u: note that none of the vertices from S^* is removed. By the definition of new rewards, we have $a_v - a_u/n^2 < \bar{a}_v \cdot a_u/n^2 \leq a_v$ for all $v \in S^*$. So $\frac{a_u}{n^2} \sum_{v \in S^*} \bar{a}_v > a(S^*) - \frac{a_u}{n} \geq (1 - 1/n)a(S^*)$, which implies (as $\bar{a}$ is integer valued) that $\bar{a}(S^*) \geq k := \lceil \frac{n^2}{a_u}(1 - \frac{1}{n})a(S^*) \rceil$. For this choice of k, the k-TSP algorithm is guaranteed to find a subset $S \subseteq V$ with $\bar{a}(S) \geq k$ and $f(S) \leq 2 \cdot f(S^*)$. The ratio of this solution is:

$$\frac{f(S)}{a(S)} \leq \frac{2f(S^*)}{a(S)} \leq \frac{n^2}{a_u} \cdot \frac{2f(S^*)}{\bar{a}(S)} \leq \frac{2}{1 - 1/n} \cdot \frac{f(S^*)}{a(S^*)}.$$

Hence the above algorithm achieves a $2 + o(1)$ approximation guarantee for minimum ratio TSP. ∎

Let $\gamma = 2 + o(1)$ denote the approximation guarantee for minimum ratio TSP from Theorem 4.2. We now show that this algorithm leads to an approximate separation oracle.

Lemma 4.3 *There is a polynomial time algorithm, that given vectors $\mathbf{b} = \{b_t^i : (i, t) \in \mathcal{D}\}$ and $\bar{\mathbf{b}} = \{\bar{b}_{st}^i : (i, t) \in \mathcal{D}, 1 \leq s \leq t\}$, outputs one of the following:*

1. *A constraint in (DLP) that is violated by $(\mathbf{b}, \bar{\mathbf{b}})$.*

2. *Certificate that $(\frac{1}{\gamma}\mathbf{b}, \frac{1}{\gamma}\bar{\mathbf{b}})$ is feasible in (DLP).*

Proof: The first set of constraints in (DLP) and non-negativity of $\bar{b}_{st}^i$ are easy to verify since they are polynomial in number. Below we assume that these are satisfied.

In order to verify the second set of constraints in (DLP), we use Theorem 4.2. For each $s \in [T]$ define an instance $\mathcal{I}_s$ of minimum ratio TSP on metric (V, w), depot r and rewards $a_v := \sum_{t=s}^{T} \bar{b}_{st}^v$ for all $v \in V$. Let A_s denote the solution found by the approximation algorithm of Theorem 4.2 on $\mathcal{I}_s$.

If any solution A_s has ratio less than one then it provides a violated constraint for $(\mathbf{b}, \bar{\mathbf{b}})$. This corresponds to the first condition in the lemma.

If every solution A_s has ratio at least one, we will show that the second condition in the lemma holds. The non-negativity constraints are clearly satisfied by $(\frac{1}{\gamma}\mathbf{b}, \frac{1}{\gamma}\bar{\mathbf{b}})$. To check the first set of constraints in (DLP), note that for any $(i, t) \in \mathcal{D}$ and $s \in [T]$,

$$\frac{1}{\gamma}\left(b_t^i - \bar{b}_{st}^i\right) \leq \max\left\{0, b_t^i - \bar{b}_{st}^i\right\} \leq H_{st}^i.$$

13

To check the second set of constraints, note that for any $s \in [T]$ we have by the approximation guarantee in Theorem 4.2,

$$\min_{S \subseteq V} \frac{f(S)}{\sum_{v \in S} \sum_{t=s}^{T} \bar{b}_{st}^v} \geq \frac{1}{\gamma}.$$

This implies that $(\frac{1}{\gamma}\mathbf{b}, \frac{1}{\gamma}\bar{\mathbf{b}})$ satisfies all these constraints. ∎

Using the above separation oracle for (DLP) within the ellipsoid algorithm, we obtain a γ-approximately optimal solution to (DLP), see eg. [20]. Then solving (LP) restricted to the (polynomially many) variables that are dual to the constraints generated in solving (DLP), we obtain a γ-approximately optimal solution to (LP) as well.

Running Time. Using the linear programming algorithms in [29, 30] along with some preprocessing, the running time of the above approach is dominated by $\tilde{O}(\mathcal{D}^3 T^3)$ plus $\tilde{O}(\mathcal{D}T^2)$ calls[1] to a subroutine for:

- submodular function minimization in case of JRP.

- minimum-ratio TSP in case of IRP.

5 Concluding Remarks

We presented an $\mathcal{O}\left(\frac{\log T}{\log \log T}\right)$-approximation algorithm for the submodular-JRP and the IRP when holding costs are polynomial functions. Moreover, this approach applies to any ordering cost for which the correponding LP relaxation can be solved approximately and the ordering cost satisfies an approximate notion of fractional subadditivity. Obtaining a constant-factor approximation algorithm for submodular JRP and IRP in general metrics (even with linear holding costs) remain the main open questions.

Acknowledgments

References

[1] Aksoy, Y., S. S. Erenguc. 1993. Multi-item inventory models with co-ordinated replenishments: a survey. *International Journal of Operations & Production Management* **8**(1) 63–73.

[2] Anily, S., A. Federgruen. 1990. One warehouse multiple retailer systems with vehicle routing costs. *Management Science* **36**(1) 92–114.

[3] Anily, S., A. Federgruen. 1993. Two-echelon distribution systems with vehicle routing costs and central inventories. *Operations Research* **41**(1) 37–47.

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

[5] Bertazzi, L. 2008. Analysis of direct shipping policies in an inventory-routing problem with discrete shipping times. *Management Science* **54**(4) 748–762.

[6] Bienkowski, M., J. Byrka, M. Chrobak, L. Jez, J. Sgall. 2013. Better approximation bounds for the joint replenishment problem. *arXiv preprint arXiv:1307.2531* .

---

[1]The $\tilde{O}$ notation hides logarithmic factors.

[7] Burns, L. D., R. W. Hall, D. E. Blumenfeld, C. F. Daganzo. 1985. Distribution strategies that minimize transportation and inventory costs. *Operations Research* **33**(3) 469–490.

[8] Campbell, A. M., M. W. P. Savelsbergh. 2004. A decomposition approach for the inventory-routing problem. *Transportation Science* **38**(4) 488–502.

[9] Chan, L. M. A., A. Federgruen, D. Simchi-Levi. 1998. Probabilistic analyses and practical algorithms for inventory-routing models. *Operations Research* **46**(1) 96–106.

[10] Cheung, M., A. Elmachtoub, R. Levi, D. B. Shmoys. 2015. The submodular joint replenishment problem. *Mathematical Programming* 1–27doi:10.1007/s10107-015-0920-3.

[11] Chien, T. W., A. Balakrishnan, R. T. Wong. 1989. An integrated inventory allocation and vehicle routing problem. *Transportation Science* **23**(2) 67–76.

[12] Coelho, L. C., J.-F. Cordeau, G. Laporte. 2014. Thirty years of inventory routing. *Transportation Science* **48**(1) 1–19.

[13] Federgruen, A., Y. S. Zheng. 1992. The joint replenishment problem with general joint cost structures. *Operations Research* **40**(2) 384–403.

[14] Federgruen, A., P. Zipkin. 1984. A combined vehicle routing and inventory allocation problem. *Operations Research* **32**(5) 1019–1037.

[15] Feige, U. 1998. A Threshold of Ln N for Approximating Set Cover. *J. ACM* **45**(4) 634–652.

[16] Feige, U. 2006. On maximizing welfare when utility functions are subadditive. *Proceedings of the Thirty-eighth Annual ACM Symposium on Theory of Computing*. STOC '06, ACM, New York, NY, USA, 41–50.

[17] Fukunaga, T., A. Nikzad, R. Ravi. 2014. Deliver or hold: Approximation algorithms for the periodic inventory routing problem. *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, AP-PROX/RANDOM 2014, September 4-6, 2014, Barcelona, Spain*. 209–225.

[18] Garg, N. 2005. Saving an epsilon: A 2-approximation for the k-mst problem in graphs. *Proceedings of the Thirty-seventh Annual ACM Symposium on Theory of Computing*. STOC '05, ACM, New York, NY, USA, 396–402.

[19] Jain, K., M. Mahdian, E. Markakis, A. Saberi, V. V. Vazirani. 2003. Greedy facility location algorithms analyzed using dual fitting with factor-revealing LP. *J. ACM* **50**(6) 795–824.

[20] Jansen, K. 2003. Approximate strong separation with application in fractional graph coloring and preemptive scheduling. *Theor. Comput. Sci.* **302**(1-3) 239–256.

[21] Joneja, D. 1990. The joint replenishment problem: new heuristics and worst case performance bounds. *Operations Research* **38**(4) 711–723.

[22] Levi, R., R. Roundy, D. Shmoys, M. Sviridenko. 2008. A Constant Approximation Algorithm for the One-Warehouse Multiretailer Problem. *Management Science* **54**(4) 763.

[23] Levi, R., R. O. Roundy, D. B. Shmoys. 2006. Primal-Dual Algorithms for Deterministic Inventory Problems. *Mathematics of Operations Research* **31**(2) 267–284.

[24] Nonner, T., M. Sviridenko. 2013. An efficient polynomial-time approximation scheme for the joint replenishment problem. *Integer Programming and Combinatorial Optimization*. Springer, 314–323.

[25] Queyranne, M. 1986. A polynomial-time, submodular extension to roundy's 98% effective heuristic for production/inventory. *Robotics and Automation. Proceedings. 1986 IEEE International Conference on*, vol. 3. IEEE, 1640–1640.

[26] Schrijver, A. 2003. *Combinatorial Optimization*. Springer-Verlag Berlin Heidelberg.

[27] Shmoys, D. B., D. P. Williamson. 1990. Analyzing the held-karp tsp bound: a monotonicity property with application. *Information Processing Letters* **35**(6) 281 – 285.

[28] Teo, C. P., D. Bertsimas. 2001. Multistage lot sizing problems via randomized rounding. *Operations Research* 599–608.

[29] Vaidya, P. M. 1990. An Algorithm for Linear Programming which Requires O((($m+n$)n^2 + ($m+n$)$^{1.5}$n)L) Arithmetic Operations. *Mathematical Programming* **47** 175–201.

[30] Vaidya, P. M. 1996. A new algorithm for minimizing convex functions over convex sets. *Mathematical Programming* **73** 291–341.

[31] Viswanathan, S., K. Mathur. 1997. Integrating routing and inventory decisions in one-warehouse multiretailer multi-product distribution systems. *Management Science* **43**(3) 294–312.

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

[33] Wolsey, L. A. 1980. Heuristic analysis, linear programming and branch and bound. *Mathematical Programming Studies* **13** 121–134.