Deliver or Hold: Approximation Algorithms for the Periodic Inventory Routing Problem

Takuro Fukunaga[1], Afshin Nikzad[2], and R. Ravi[3]

1 **National Institute of Informatics, Japan**
JST, ERATO, Kawarabayashi Large Graph Project, Japan
takuro@nii.ac.jp

2 **MS&E Department, Stanford University, USA**
nikzad@stanford.edu

3 **Tepper School of Business, Carnegie Mellon University, USA**
ravi@cmu.edu

Abstract

The inventory routing problem involves trading off inventory holding costs at client locations with vehicle routing costs to deliver frequently from a single central depot to meet deterministic client demands over a finite planing horizon. In this paper, we consider periodic solutions that visit clients in one of several specified frequencies, and focus on the case when the frequencies of visiting nodes are nested. We give the first constant-factor approximation algorithms for designing optimum nested periodic schedules for the problem with no limit on vehicle capacities by simple reductions to prize-collecting network design problems. For instance, we present a 2.55-approximation algorithm for the minimum-cost nested periodic schedule where the vehicle routes are modeled as minimum Steiner trees. We also show a general reduction from the capacitated problem where all vehicles have the same capacity to the uncapacitated version with a slight loss in performance. This reduction gives a 4.55-approximation for the capacitated problem. In addition, we prove several structural results relating the values of optimal policies of various types.

1998 ACM Subject Classification G.1.6 Optimization

Keywords and phrases Inventry Routing Problem, Approximation algorithm, Prize-collecting Steiner Tree

Digital Object Identifier 10.4230/LIPIcs.APPROX-RANDOM.2014.209

1 Introduction

The Inventory Routing Problem (IRP) is a classical problem in the Supply Chain Optimization area of the Operations Management literature [7, 13, 18, 19, 21] that captures the trade-off between the holding costs for inventory and the routing costs of replenishing the inventory at various locations in a supply chain. It arises in the context of vendor-managed inventory systems where the supplier running a depot manages the inventory at its client demand locations [30]. The general problem involves multiple products that are stocked at multiple depots, that must be shipped to meet the demand for these products arising at multiple locations (clients) specified over the course of a planning horizon that involves several time periods (days or rounds). The costs of holding a unit of each product per day at each of the clients are specified to compute the inventory holding costs; vehicles are available at the depots with given capacities and transportation costs in the metric defined by the depots and clients determine the vehicle routing costs. The goal of the problem is to find a set of vehicle

17th Int'l Workshop on Approximation Algorithms for Combinatorial Optimization Problems (APPROX'14) / 18th Int'l Workshop on Randomization and Computation (RANDOM'14).
Editors: Klaus Jansen, José Rolim, Nikhil Devanur, and Cristopher Moore; pp. 209–225
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

routes for each day of the planning horizon that delivers sufficient units of products at each location to satisfy all demands, and minimizes the total sum of inventory *holding* costs and vehicle *routing* costs over the whole planning horizon. Note that we are not allowed to have any demand backlog at any client location.

Being a natural extension of the classical inventory holding problem to a networked setting, many of the early papers try to adopt the approach from the inventory literature of looking for *policies* under a specified random pattern of demand at the locations [2, 3] that optimize the infinite-horizon costs. However, no optimal or near-optimal policies are known in the general case of the problem. The more tactical solutions to this real-life problem involve focusing on a finite planning horizon as we have formulated it above: in this given horizon, the demands can be assumed to be deterministic (realizations of some underlying random process or very good forecasts of them). The problem can then be cast as an integer program [14] that decides on the vehicle routes per day and the amounts delivered per day. Solution approaches for this formulation typically involve heuristics [15, 17] that try to manage the short-term formulation with extra constraints or objectives to take care of extending the solutions to the next finite planning horizon overlapping with the current one.

In this paper, we address the most-studied version of the finite horizon problem with a *single depot*. We focus on the tactical version of the problem where the demands at the clients are assumed to be deterministic over the horizon and design constant-factor approximation algorithms for periodic solutions which we define next.

Periodic Schedules

Even though the single-depot version of the IRP can be cast as an IP, finding solutions that serve the clients at arbitrarily spaced time periods can be practically cumbersome to implement. Indeed, Anily and Federgruen noted early on [2]: “The complexity of the structure of optimal policies makes them difficult, if not impossible, to implement even if they could be computed efficiently". A very natural restriction that has been considered is to require that every client is visited according to a *periodic* schedule, i.e., once every f days for some frequency f associated with that client. A simple example is a delivery schedule that visits a client in a particular day of the week.

Another common further restriction on the set of frequencies used is that longer frequencies are multiples of all the smaller frequencies: e.g., clients can be visited once every week, or once every month (4 weeks) or once every quarter (every 12 weeks), so that in days, the periods are 7, 28 and 84. Such schedules are called *nested periodic* [31]. A particular class of such policies where the frequencies are powers of two is very well studied in Inventory Management following the seminal work of Roundy [27] on the efficacy of power-of-two policies as very good approximations for more complicated replenishment policies. Our main result is a constant-factor approximation algorithm for nested periodic schedules, that also gives corollaries for power-of-two schedules as well as for general power-of-k schedules for $k > 2$ (We call these 2-periodic and k-periodic schedules in the sequel).

Partition Schedules

While we require each node to be visited with a frequency that is a power of two in the 2-periodic schedules, we have a choice of the vehicle route used in round 2 versus round 4. In the general problem, we may use two different vehicle routes on days 2 and 4 since the second route will also include nodes visited with frequency 4 in addition to those visited with frequency 2: we call such schedules simply 2-periodic schedules to denote their generality.

However, we may further restrict that the vehicle route used to visit the clients of frequency 2 must always be the same; in this case, in round 4, we would have two different vehicle routes, with one visiting all nodes of frequency 2 and the other visiting nodes of frequency 4. Note that we have effectively partitioned the clients into subsets based on their frequencies and the schedule specifies a vehicle route that visits each set independent of the others. Such schedules are called *partition schedules* or policies [3]; we will term a partition schedule for the general nested case as a nested partition policy and the power-of-two version as a 2-partition policy. Note that partition schedules are also periodic.

2 Our Results

2.1 Problem Definition

In the IRP, we are given a complete graph G with a node set $V(G)$ and an edge set $E(G)$, and a metric distance function w (i.e., $w(ab) + w(bc) \geq w(ac)$ for any nodes $a, b, c \in V(G)$). A single depot node (or root) r is specified as well as a set of client nodes. The depot has infinite production capacity, which means there are as many units of the production as we need available at the depot.

We are given a planning horizon of length $T + 1$ (our rounds are numbered 0 through T) and are asked to satisfy the demands of the clients in these $T + 1$ rounds. Thus, client i, which has a demand d_i^j in the j-th round of the planning horizon, should be satisfied by the depot using a set of (possibly capacitated) vehicles as follows: In each round, for each vehicle, we determine a sequence of some nodes of $V(G)$, and the vehicle visits those nodes in that order. The vehicle starts at the depot node, picks up the products, and then visits client nodes, to deliver a portion of its shipment to the client and satisfy the client's demands until the next visit; in the capacitated version, the vehicle cannot carry more units than its capacity C. In this paper, we assume uniform capacities for all the vehicles. Also, we assume that the client demands are discrete, and that partial (fractional) satisfaction of a unit of demand is not allowed.

Assume inventory is used in the order it was delivered to satisfy demand. For each unit of the production that client i receives in round s and keeps in its inventory till round t, it incurs a holding cost $h^i_{s,t}$. Hence the client i is visited in rounds $0, t_1, \ldots, t_k$, then the holding cost for i is $\sum_{j=0}^{k} \sum_{s=t_j}^{t_{j+1}} d_i^s h^i_{t_j s}$, where we let $t_0 = 0$ and $t_{k+1} = T + 1$. Note the general structure used for modeling the holding costs. The only assumption we need for the holding costs is monotonicity, i.e. for any client i and any three rounds $u \leq s \leq t$, we need to have $h^i_{ut} \geq h^i_{st}$.

For a given solution x, the sum of all the holding costs incurred by all the clients over the whole horizon is called the *holding cost* $h(x)$. Similarly, the sum of the distances traveled by all the vehicles in all the rounds is called the *shipping cost* $w(x)$. The cost of a solution x to the IRP, $c(x)$, is the sum of its holding cost and its shipping cost. We want to determine the solution (also called schedule or policy), that specifies vehicle routes and demand deliveries in each route, such that the total incurred cost is minimized. Note that for feasibility, the amount at any location at any period is at least its demand up to now.

As noted above, we consider only periodic policies in this paper. A frequency is defined as a positive integer, and we assume that available frequencies are given. In periodic policies, a solution assigns a frequency to each client node. If a client node is assigned frequency f, then the vehicle in round pf has to visit the node for each positive integer p up to $\lfloor T/f \rfloor$. Partition policies are different from periodic policies only in the fact that a client node is visited in the same route every time.

Let $q_0, q_1, \ldots, q_k$ be integers such that $q_0 = 1$ and $q_j \geq 2$ for $j = 1, 2, \ldots, k$. In a nested (periodic or partition) policy, the available frequencies are $f_0, f_1, \ldots, f_{k+1}$ that are defined as $f_i = \prod_{j=0}^{i} q_j$ for $i = 0, 1, \ldots, k$, and $f_{k+1} = +\infty$. We need f_{k+1} to represent nodes visited only once in round 0. A k'-periodic policy denotes a policy with frequencies defined by $q_j = k'$ for each $j = 1, 2, \ldots, k$.

Summing up, the problem considered in this paper is summarized as follows.

Problem IRP

Input:
- A metric (G, w) with a depot node $r \in V(G)$.
- A planning horizon length $T + 1$.
- A demand d_i^j for each $i \in V(G) \setminus \{r\}$ and $j \in \{0, 1, \ldots, T\}$.
- A holding cost $h_{s,t}^i$ for each $i \in V(G) \setminus \{r\}$ and $s, t \in \{0, 1, \ldots, T\}$.
- A vehicle capacity C in capacitated case.
- Available frequencies $f_0, f_1, \ldots, f_{k+1}$, where $f_{k+1} = +\infty$. The nested policy defines f_i from $q_0, q_1, \ldots, q_k$ as $\prod_{j=0}^{i} q_j$ for $i = 0, 1, \ldots, k$.

Output:
- Allocation of a frequency to each client.
- A set of vehicle routes for each round.

Objective: Minimize the sum of the holding cost and the shipping cost.

2.2 Techniques

We present constant-factor approximation algorithms for designing periodic policies for the general capacitated version of the IRP. Our main contribution is a simple reduction of the problem to a carefully designed instance of a prize-collecting vehicle routing problem (PCVRP), where vehicles from a root node (depot) must either visit each client or pay a pre-specified penalty for not visiting it, with the objective function summing the cost of the route and the penalties of uncovered clients. The prize-collecting TSP [6] as well as the prize-collecting Steiner tree problems have been well studied [10, 22]; the best-known performance ratios for these problems are some constants slightly below 2 [4].

We describe the key idea behind our algorithm for the uncapacitated case where the vehicles can carry any number of units in each round: our reduction uses roughly one copy of the metric for each frequency, where the edge lengths in the metric are scaled appropriately by the number of times the vehicle route of the corresponding frequency will be used in the overall IRP schedule: if the planning horizon is T rounds, and the frequency represented by the i^{th} copy is f_i, this scale factor is roughly T/f_i. The penalties of a node in each of the copies are carefully set so that if a node is covered by a vehicle route for the first time in the i^{th} copy, then the sum of the penalties of the first $(i-1)$ copies gives the holding cost for this node if it were visited only every f_i rounds. In this way, the two components of the PCVRP problem - covering with routes and penalties - capture the two components of the IRP - routing and inventory holding costs.

We can observe that periodic schedules in the original IRP instance correspond to solutions with some monotonicity property in the PCVRP instance. Hence we can save the approximation factor incurred by transforming PCVRP solutions into periodic schedules if we can compute monotone PCVRP solutions. To do this, we solve a natural linear programming relaxation of the problem strengthened by adding the constraint on monotonicity, and then use randomized threshold rounding for converting fractional solutions to an integral solution. This gives rise to the approximation guarantees that we provide.

We can extend this idea to the capacitated case. When all vehicles have the same capacity, then we can relate the cost of the corresponding capacitated VRP to a lower-bound defined from the shortest length from the root to each client scaled by the ratio of its demands to the capacity. To solve the overall problem, we solve the uncapacitated version of the problem by ignoring the capacity, and break the obtained routes for satisfying the capacity constraints. The key idea is that the costs for connecting the generated sub-routes to the root is bounded by the lower-bound we obtain.

2.3 Summary of Results

For simplicity, in the sequel, we consider the cost of the vehicle route at any period to be modeled by the length of a minimum-cost Steiner tree connecting the depot to the clients that are visited in that period. Our results easily generalize to the case when the vehicle routes are TSP tours over the same set with the appropriate replacement of Steiner tree with TSP in the guarantees below.

A solution x specifies the Steiner trees used by the vehicles at every round, and hence delivers at every visited client, as many units as are needed to satisfy the demand from the current round to the next round when it is visited in the solution. We denote the total holding cost incurred at all clients by the solution x by $h(x)$ and the total routing cost of all vehicles in x by $w(x)$. Let ρ_{PC} denote the best-known approximation factor for the prize-collecting Steiner tree problem (PCST) - the value of ρ_{PC} is currently $(2-\epsilon)$ for some small constant $\epsilon > 0$ [4].

Constant Approximation for Uncapacitated IRP

First we present constant-factor approximation algorithms for uncapacitated IRPs, which are summarized as follows.

1. We design a 2.55-approximation for the minimum-cost nested periodic policy for the uncapacitated IRP.
2. We give an $\alpha_{\mathsf{NE}}\rho_{\mathsf{PC}}$-approximation algorithms for the minimum-cost nested partition and periodic policies for the uncapacitated IRP where α_{NE} is a constant not larger than 2 which is defined in terms of the given frequencies.
3. We design an $\alpha_{\mathsf{NA}}\rho_{\mathsf{PC}}$-approximation algorithm for the minimum-cost nested partition policy for the uncapacitated IRP where α_{NA} is a constant not larger than 2. Note that α_{NA} is a different constant from α_{NE}.
4. We also study general partition policies for uncapacitated IRP which do not need to be nested and provide a $4\rho_{\mathsf{PC}}$-approximation algorithm for the minimum-cost general partition policies. This result is a consequence of the above $\alpha_{\mathsf{NA}}\rho_{\mathsf{PC}}$-approximation for 2-partition policies and a structural relation between 2-partition policies and general partition policies.

In this article, because of the space limitation, we only explain the 2.55-approximation algorithm for the nested periodic policies (Corollary 5) and the $\alpha_{\mathsf{NE}}\rho_{\mathsf{PC}}$-approximation algorithms for the nested partition policies (Theorem 6) in Section 5.

Constant Approximation for IRP with Uniform Capacities

Next we show a general reduction of the capacitated IRP to the corresponding uncapacitated version with a slightly worse guarantee (details are in Section 6).

1. We show that for *any* periodic policy for the IRP, a γ-approximation algorithm for the uncapacitated version can be used to design a $(\gamma+2)$-approximation for the corresponding capacitated version (Theorem 9).
2. For *any* partition policy, we show that a γ-approximation algorithm for the uncapacitated version gives a $(\gamma + 4)$-approximation when each client has a constant demand rate (Theorem 10).

As corollaries, these results yield a counterpart for every result for uncapacitated IRP.

Structural Results – Relations Between Various Types of Policies

Finally we study relations between partition policies and periodic policies. We also relate the optimal value of *any* periodic schedule to that of an optimal 2-periodic schedule for the same instance in the spirit of the result of Roundy [27].

1. We provide an upper bound on the cost of the optimum nested partition policy in terms of the cost of the optimum nested periodic policy for uncapacitated IRP. Formally, we show that $c(x^*_{\mathsf{NE}}) \leq c(x^*_{\mathsf{NA}}) \leq \alpha_{\mathsf{NE}} c(x^*_{\mathsf{NE}})$, where x^*_{NE} and x^*_{NA} respectively denote the optimum nested periodic and partition policies. As a consequence of this result, any ϕ-approximation algorithm for the minimum-cost nested partition policy for uncapacitated IRP is also an $\alpha_{\mathsf{NE}}\phi$-approximation for the corresponding minimum-cost nested periodic policy.
2. We show that for any partition policy x to the uncapacitated IRP, there exists a 2-partition policy y such that $w(y) \leq 2w(x)$ and $h(y) \leq h(x)$. Note that this immediately implies a constant factor approximation algorithm for optimal partition policies with arbitrary frequencies.
3. We do not have an analogous result for arbitrary periodic policies. However, for any k-periodic policy x for $k \geq 3$, we show that there exists a 2-periodic policy y such that $w(y) \leq 2w(x)$ and $h(y) \leq h(x)$.

We do not prove these results in this article due to the space limiatation. We recommend referring to the full version.

2.4 Roadmap

We briefly survey related work in Section 3. Then, in order to illustrate the idea of our reduction from the uncapacitated IRP to the PCVRP, we present an approximation algorithm for the uncapacitated version of 2-periodic or power-of-two policies in Section 4. We present the approximation algorithms for nested policies in uncapacitated IRP in Section 5, and the reductions from the capacitated to the uncapacitated IRP in Section 6. We conclude the paper in Section 7.

3 Related Work

IRP has a vast literature that is addressed in several surveys [20, 11, 28, 29] that focus on different variants of the problem such as those considering stochastic demands, capacitated vehicles or capacities on local client inventories. Solution approaches for these versions in turn can be categorized into three main groups: (i) designing heuristics, (ii) designing policies typically for the infinite-horizon stochastic demand version and showing their (near-)optimality, and most directly related to our work, (iii) designing approximation algorithms for special cases of the problem. We only review the last stream below.

We can model the Vehicle Routing Problem with vehicle capacity constraints by an IRP instance in which we are given a single round and a capacitated vehicle. When the clients demands and the vehicle capacities are uniform, Charikar, Khuller, and Raghavachari [16] gave a 5-approximation. In addition to the VRPs, inventory replenishment problems such as the Joint Replenishment Problem (JRP) can also be seen as a special case of the IRP. In the JRP, there is no metric or clients but several retailers, and each retailer has a local retailer-ordering cost that must be paid whenever it places an order. In addition, there is a warehouse ordering cost that must be paid whenever any retailer places an order. These fixed costs for ordering at any time period can be represented by a simple two level tree: the edge from the root (warehouse) to a dummy node represents the warehouse order cost; edges from the dummy to each client/retailer represents the retailer order cost. A subset of retailers ordering at a round involves paying the cost of the tree induced by them and the root converting this to a case of the IRP on a star network. For the JRP, the first constant approximation ratio was provided by Levi, Round, and Shmoys [25] by giving a 2-approximation algorithm. Later, in [26], they improved the approximation ratio to 1.8. The problem is also studied in the online setting, where the demands are not given in advance, but they arrive online. For this case, a 3-approximation algorithm was provided in [12]. While our work generalizes the JRP to arbitrary metrics and still derives constant-factor approximation algorithms, we focus only on periodic schedules.

4 2-Periodic Policies for Uncapacitated IRP

4.1 Preliminaries

In this section we modify the objective function slightly to remove the contribution of the routing cost in the initial round to simply future calculations.

Note that in round 0 in any partition or periodic policy, all of the client nodes should be visited assuming that there is nonnegative demand in that round (so as to satisfy these demands). We therefore assume without loss of generality that any partition or periodic policy uses an (approximately) optimum Steiner tree in round 0 to visit all of the client nodes.

Let x^*_{PE} denote the optimum 2-periodic policy. Given any weight function $\psi\colon E(G) \to \mathbb{R}$ and a sub-graph H of G, we define $\psi(H)$ to be $\sum_{e \in E(H)} \psi(e)$.

For any policy x, let $h(x)$ and $\bar{w}(x)$ respectively denote the holding and shipping cost incurred in x. Define $\bar{c}(x)$ to be the total cost of the policy, i.e. $\bar{c}(x) = h(x) + \bar{w}(x)$. For any policy x, let $w_0(x)$ be the incurred shipping cost in round 0 of x and define $w(x) = \bar{w}(x) - w_0(x)$ and $c(x) = \bar{c}(x) - w_0(x)$. To simplify the analysis, we define the approximation factor in terms of the refined cost function $c(x)$ (Note that this does not worsen the performance factors).

4.2 Main Idea of the Algorithm

We set up an instance of PCST and solve this instance using an existing approximation algorithm for PCST, e.g. Goemans-Williamson Algorithm (denoted by the GW Algorithm from now on) [22]. Then, using the solution, we construct a policy for the originally given IRP instance.

Here we formally define the PCST instance. Let $L = \lfloor \log T \rfloor$. The instance includes a dummy root node r^*, and $L+1$ copies of G, namely $G_0, \ldots, G_L$. The only difference between G_i and G is the edge weights. Recall that $w(e)$ denotes the weight of an edge e in G. Let the copy of e in G_i be denoted by e_i, and define the weight of e_i, i.e. $w(e_i)$, as follows:

$$w(e_i) = w(e) \cdot \left\lceil \frac{\lfloor T/2^i \rfloor}{2} \right\rceil. \tag{1}$$

To avoid the risk of confusion, when it is needed, we denote the weight function in the graph G by $w_G(\cdot)$, and the weight function in G_i by $w_i(\cdot)$. For convenience, we sometimes identify $S \subseteq V(G_i)$ with $\{v \in V \mid v_i \in S\}$, and $U \subseteq E(G_i)$ with $\{e \in E(G) \mid e_i \in U\}$.

Let r_i be the copy of the root node r in G_i. Vertices $r_0, \ldots, r_L$ are connected to r^* with edges of weight 0. To finish the definition of the PCST instance, it only remains to define the penalties of nodes. For any node $v \in V(G)$ denote its copy in G_i by v_i. We define the penalties $p(v_i)$ so that $\sum_{j=0}^{i-1} p(v_j)$ will be equal to the total holding cost that node v pays if it is visited with frequency 2^i. More formally, the penalties are defined as follows:

$$p(v_i) = \begin{cases} \mathcal{H}(v,1), & \text{if } i = 0 \\ \mathcal{H}(v,i+1) - \mathcal{H}(v,i), & \text{if } 0 < i \le L \end{cases} \tag{2}$$

where $\mathcal{H}(v,i)$ will be defined below. Before that, note that with this definition of the penalties, we will have $\sum_{j=0}^{i-1} p(v_j) = \mathcal{H}(v,i)$.

As we mentioned before, we want to set $\mathcal{H}(v,i)$ so that it is equal to the total holding cost that node v pays if it is visited with frequency 2^i. To define this quantity more formally, see that when v is visited every 2^i rounds, then it will be visited in rounds $k \cdot 2^i$ for all integers k such that $k \le T/2^i$. Hence, when visiting node v in round $k \cdot 2^i$, we should deliver the demand it requires from round $k \cdot 2^i$ to round $\min\{T, (k+1) \cdot 2^i - 1\}$. This will determine the holding cost that v incurs for period $k \cdot 2^i$ to $\min\{T, (k+1) \cdot 2^i - 1\}$. Summing this over all $k \le T/2^i$ will give $\mathcal{H}(v,i)$, i.e. the total holding cost incurred by v when it is visited every 2^i rounds. Thus, from (2) we get the following proposition:

▶ **Proposition 1.** *For any $i \le L$ we have $\sum_{j=0}^{i-1} p(v_j) = \mathcal{H}(v,i)$.*

▶ **Proposition 2.** *For any i from 0 to L we have $p(v_i) \ge 0$.*

4.3 2-Periodic Policies

For finding a 2-periodic policy, we need to assign a frequency 2^i to each client node where i can vary from 0 to L. If a node is assigned frequency 2^i, then the policy guarantees to visit it every 2^i rounds and delivers the required demand until the next visit. Note that unlike the partition policy, the node may be reached via the different routes (trees) in every visit. To define a periodic policy completely, we need to define these routes as well.

Let S_i be the set of client nodes that are assigned frequency 2^i. For any round j, the nodes in S_i need to be visited in that round if j is a multiple of 2^i. In other words, if we define $\xi(j)$ to be the largest integer k such that j is a multiple of 2^k, then the nodes in $S_0, \ldots, S_{\xi(j)}$ need to be visited in round j. Consequently, for any round j, the shipping routes are defined by a tree $T_{\xi(j)}$ which visits the nodes in $S_0, \ldots, S_{\xi(j)}$. Below we present an algorithm which provides an approximately optimum 2-periodic policy by outputting the node sets $S_0, \ldots, S_L$, along with the set of trees $T_0, \ldots, T_L$. Again, note that by the definition, the tree T_i visits the nodes in $S_0, \ldots, S_i$.

Algorithm *Periodic Policy*
Input: An IRP instance
Output: A 2-periodic policy defined by subsets $S_0, \dots, S_L$ of the client node set and trees $T_0, \dots, T_L$
1. **for** $i = 0$ **to** L
2. **do** $S_i \leftarrow \emptyset$
3. Construct the PCST instance.
4. Solve the PCST instance.
5. For all i, let Q_i be the set of client nodes in G_i which, using the tree U_i, got connected to r^* in the solution.
6. **for** $i = 0$ **to** L
7. **do** $S_i \leftarrow Q_i \setminus \cup_{j=0}^{i-1} Q_j$
8. Let H_i be the subgraph of G with the edge set $E(U_0) \cup \dots \cup E(U_i)$.
9. Let T_i be an arbitrary spanning tree in H_i.
10. Output $S_0, \dots, S_L$ and $T_0, \dots, T_L$.

▶ **Lemma 1.** *In any periodic policy, and for any integer i such that $i \leq L$, the tree T_i is used in exactly $\left\lceil \frac{\lfloor T/2^i \rfloor}{2} \right\rceil$ number of the rounds (recall that round 0 is excluded).*

Proof. Since for each round j we use the tree $T_{\xi(j)}$, then the number of times that the tree T_i is used is equal to the number of integers p such that $1 \leq p \leq T$ and $p/2^i$ is an odd integer. It is easy to verify that there are exactly $\left\lceil \frac{\lfloor T/2^i \rfloor}{2} \right\rceil$ values which can be assigned to p. ◀

▶ **Theorem 2.** *If Step 4 of Algorithm Periodic Policy uses a ρ_{ST}-approximation algorithm for solving the PCST instance, then the algorithm achieves approximation factor $2\rho_{\mathsf{PC}}$. If the GW algorithm is used instead of the ρ_{ST}-approximation algorithm, then Algorithm Periodic Policy finds a periodic policy x_{PE} such that $c(x_{\mathsf{PE}}) \leq 2h(x^*_{\mathsf{PE}}) + 4w(x^*_{\mathsf{PE}})$.*

Proof. Let y^* denote the optimum solution for the constructed PCST instance. Let $p(y^*)$, $w(y^*)$, and $c(y^*)$ respectively denote the penalty cost, the tree cost, and the total cost of y^*. We hence have $c(y^*) = p(y^*) + w(y^*)$.

The proof consists of three steps. In the first step, we prove that $p(y^*) \leq h(x^*_{\mathsf{PE}})$ and $w(y^*) \leq w(x^*_{\mathsf{PE}})$. Then, in the second step, we observe that Step 4 of Algorithm *Periodic Policy* finds a solution $\hat{y}$ of cost at most $\rho_{\mathsf{ST}} c(y^*)$ when it uses a ρ_{ST}-approximation algorithm for PCST. In fact, this needs no proof since it follows from the definition of ρ_{ST}-approximation. When Step 4 uses the GW algorithm, then it finds a solution $\hat{y}$ of cost at most $p(y^*) + 2w(y^*)$, which was proven in [22]. Finally, in the third step, we show that Algorithm *Periodic Policy* converts $\hat{y}$ into a periodic policy x_{PE} such that $c(x_{\mathsf{PE}}) \leq 2c(\hat{y})$.

To do the first step, given x^*_{PE}, we construct a solution y for the PCST instance such that $p(y) = h(x^*_{\mathsf{PE}})$ and $w(y) = w(x^*_{\mathsf{PE}})$. Let $S^*_0, \dots, S^*_L$ be the subsets of the client node set in the periodic policy x^*_{PE}, and let $T^*_0, \dots, T^*_L$ respectively be their associated trees. Then, construct y as follows: for each copy G_i in the PCST instance, visit the nodes in $\cup_{j=0}^{i} S^*_j$ using the tree T^*_i, and pay the penalty for the nodes in $V(G_i) \setminus (\cup_{j=0}^{i} S^*_j)$. Observe that every node $v \in S^*_i$ is visited in all of the copies except $G_0, \dots, G_{i-1}$. We have $\sum_{j=0}^{i-1} p(v_j) = \mathcal{H}(v, i)$ by Proposition 1. These mean that the overall holding cost paid for v in x^*_{PE} is equal to the overall penalty paid for (the copies of) v in y. Therefore $p(y) = h(x^*_{\mathsf{PE}})$ holds.

It is easy to see that $w(y) = w(x^*_{\mathsf{PE}})$ holds as well. Just observe that the number of times that the tree T^*_i is used in x^*_{PE} is equal to $\left\lceil \frac{\lfloor T/2^i \rfloor}{2} \right\rceil$ by Lemma 1, and hence the total tree cost paid for using T_i^* in x_{PE}^* is $w_G(T_i^*) \cdot \left\lceil \frac{\lfloor T/2^i \rfloor}{2} \right\rceil$, which is exactly equal to $w_i(T_i)$, i.e. the tree cost incurred in y for copy G_i. Summing over all i implies that $w(y) = w(x_{\mathsf{PE}}^*)$ holds.

In the last step of the proof, we show that $c(x_{\mathsf{PE}}) \leq 2c(\hat{y})$ by showing that $h(x_{\mathsf{PE}}) \leq p(\hat{y})$ and $w(x_{\mathsf{PE}}) \leq 2w(\hat{y})$. To prove $h(x_{\mathsf{PE}}) \leq p(\hat{y})$, fix a node v and let i be the smallest integer such that v_i is connected to r^* in $\hat{y}$. Hence the overall penalty paid for (the copies of) v would be at least $\mathcal{H}(v,i)$ by Lemma 1. On the other hand, by the choice of S_i in Algorithm *Periodic Policy*, we have $v \in S_i$. This guarantees that the overall holding cost paid for v in x_{PE} is exactly $\mathcal{H}(v,i)$. By the two latter facts, the overall holding cost paid for v in x_{PE} is at most the overall penalty paid for v in $\hat{y}$. By summing over all v, we get $h(x_{\mathsf{PE}}) \leq p(\hat{y})$.

It remains to show that $w(x_{\mathsf{PE}}) \leq 2w(\hat{y})$. By the choice of $T_0, \ldots, T_L$ in Algorithm *Periodic Policy*,

$$w(x_{\mathsf{PE}}) \leq \sum_{i=0}^{L} \sum_{j=i}^{L} w_j(U_i). \tag{3}$$

Now, if for any fixed i, we show that $\sum_{j=i+1}^{L} w_j(U_i) \leq w_i(U_i)$, then by (3) we have

$$w(x_{\mathsf{PE}}) \leq \sum_{i=0}^{L} \sum_{j=i}^{L} w_j(U_i) \leq \sum_{i=0}^{L} 2w_i(U_i) = 2w(\hat{y}) \tag{4}$$

where the equality in (4) is due to the fact that $\sum_{i=0}^{L} w_i(U_i) = w(\hat{y})$. Thus it only remains to show that $\sum_{j=i+1}^{L} w_j(U_i) \leq w_i(U_i)$. Equivalently, by the definition of $w_i(\cdot)$, we have to show that

$$\sum_{j=i+1}^{L} \left\lceil \frac{\lfloor T/2^j \rfloor}{2} \right\rceil \leq \left\lceil \frac{\lfloor T/2^i \rfloor}{2} \right\rceil.$$

This inequality can be proven by elementary calculations. ◀

5 Nested Policies for Uncapacitated IRP

In this section, we generalize the context of

Section 4 from power-of-two to arbitrary nested policies. We also refine the method used in Section 4 to convert our problem to a *monotone* version of a prize-collecting VRP. In particular, we present two approximation results for nested policies, one of which is for nested periodic policies, and the other of which is for nested partition policies.

Let $q_0, q_1, \ldots, q_k$ be integers such that $q_0 = 1$ and $q_j \geq 2$ for $j = 1, 2, \ldots, k$. In a nested policy, available frequencies are $f_0, f_1, \ldots, f_{k+1}$ that are defined as $f_i = \prod_{j=0}^{i} q_j$ for $i = 0, 1, \ldots, k$, and $f_{k+1} = +\infty$.

5.1 2.55-Approximation Algorithm for Nested Periodic Policies

We here present an algorithm for nested periodic policies. Our algorithm again reduces the problem to PCST as Algorithm *Periodic Policy* in Section 4 did. In our reduction, the graph and penalties are same as before (we replace L by k); The graph is the union of $k+1$ copies $G_0, G_1, \ldots, G_k$ of G and a new node r^* connected to the copies of r by edges of weight 0; The penalty $p(v_i)$ of the i-th copy of a client node v is defined by (2) where $\mathcal{H}(v,i)$ is the total holding cost that node v pays when it is assigned frequency f_i. We define the weight

$w(e_i)$ of the i-th copy of an edge e as the one we need to pay when we use it in a tree of frequency f_i, as follows.

$$w_i(e_i) = w_G(e) \left(\left\lfloor \frac{T}{f_i} \right\rfloor - \left\lfloor \frac{T}{f_{i+1}} \right\rfloor \right). \tag{5}$$

We say that a solution for the PCST instance is *monotone* when for any client node v and for any i and j such that $1 \leq i < j \leq k$, v_j is connected to r^* if v_i is connected to r^*. In our algorithm for nested periodic policies, we have to approximate a minimum cost monotone solution for the PCST instance. We here assume that there exists a ρ-approximation algorithm for this problem. At the end of this subsection, we mention that there exists an algorithm with $\rho < 2.55$. The construction of a nested periodic policy from a monotone solution for PCST is almost same as *Periodic Policy*; A client node v is assigned frequency f_i when i is the minimum index such that v_i is connected to r^* in the monotone solution for PCST. We use the tree of the solution in G_i to visit client nodes of frequency at most f_i in round t such that f_i is the maximum frequency that divides t.

In the next theorem, we show that the modified *Periodic Policy* is a ρ-approximation algorithm. We omit its proof due to the space limitation.

▶ **Theorem 3.** *Suppose that there exists a ρ-approximation algorithm for finding a minimum cost monotone solution for the PCST instance defined above. Then the problem of fining a minimum cost nested periodic policy for uncapacitated IRP admits a ρ-approximation algorithm.*

Let us discuss algorithms for approximating minimum cost monotone solutions for PCST. The algorithm due to [4] achieves $\rho_{\mathsf{PC}} = 2 - \epsilon$ for some constant ϵ currently, and the GW algorithm [22] achieves approximation factor 2 for PCST. We do not know if these algorithms can be modified for approximating monotone solutions. What we can do here is to modify the algorithm due to an unpublished work of Goemans (refer to [23, 32]). This algorithm achieves approximation factor $1/(1 - e^{-1/2}) < 2.55$ as follows: Consider an LP relaxation of PCST which has a variable $x(e)$ for representing what fraction of an edge e is chosen in a solution, and a variable $y(v)$ for representing what fraction of a terminal v is covered by the solution; The algorithm solves the LP relaxation to obtain an optimal solution (x^*, y^*); It also chooses a threshold α uniformly at random from $[e^{-1/2}, 1]$, and let $\hat{S} = \{v \mid y^*(v) \geq \alpha\}$; The algorithm outputs a Steiner tree that connects $\hat{S}$ to the root. The LP used there still gives a lower-bound on the optimal value of our problem even if we add a new constraint

$$y(v_0) \leq y(v_1) \leq \cdots \leq y(v_k)$$

for each $v \in V$, and by this new constraint, the Steiner tree output by the algorithm is monotone. It is not difficult to verify that this Steiner tree achieves the same approximation factor as before, and we therefore have the following theorem.

▶ **Theorem 4.** *The problem of finding a minimum cost monotone PCST admits an approximation factor within* $1/(1 - e^{-1/2}) < 2.55$.

Theorems 3 and 4 gives the next corollary.

▶ **Corollary 5.** *The problem of finding a minimum cost nested periodic policy for uncapacitated IRP can be approximated within a factor of* $1/(1 - e^{-1/2}) < 2.55$.

5.2 $\alpha_{\mathsf{NE}}\rho_{\mathsf{PC}}$-Approximation Algorithm for Nested Partition Policies

For approximating nested partition policies, we use the same reduction to PCST as for nested periodic policies. When we solve the constructed instance of PCST, we do not have to restrict solutions to monotone solutions here, and hence we can use a ρ_{PC}-approximation algorithm.

Let y be a solution for PCST computed by the ρ_{PC}-approximation algorithm. Let T_i be the subtree of y in G_i. While T_i is used $\lfloor T/f_i \rfloor - \lfloor T/f_{i+1} \rfloor$ times for visiting $S_0 \cup S_1 \cup \cdots \cup S_i$ in nested periodic policies, our nested partition policy uses it $\lfloor T/f_i \rfloor$ times for visiting only S_i because a client node in S_i has to be reached via the same tree at every visit.

Let

$$\alpha_{\mathsf{NE}} = \max_{1 \le i \le k} \left(1 + \frac{1}{q_i - 1}\right). \tag{6}$$

We always have $\alpha_{\mathsf{NE}} \le 2$ since $q_i \ge 2$ for $i = 1, 2, \ldots, k$.

▶ **Theorem 6.** *Suppose that there exists a ρ_{PC}-approximation algorithm for PCST. Then we can compute a nested partition policy x_{NA} such that $h(x_{\mathsf{NA}}) + w(x_{\mathsf{NA}}) \le \rho_{\mathsf{PC}} h(x^*_{\mathsf{NE}}) + \alpha_{\mathsf{NE}}\rho_{\mathsf{PC}} w(x^*_{\mathsf{NE}})$. In particular, the problem of finding a minimum cost nested partition policy for uncapacitated IRP admits an $\alpha_{\mathsf{NE}}\rho_{\mathsf{PC}}$-approximation algorithm.*

Proof. Let y be a ρ_{PC}-approximate solution for the PCST instance. Let x_{NA} be the nested partition policy computed from y by our algorithm. We compare x_{NA} with an optimal nested periodic policy x^*_{NE}. This is enough because the minimum cost of nested periodic policies is at most that of nested partition policies.

In the proof of Theorem 3, we have already proven that there exists a monotone solution $\hat{y}$ for PCST such that $w(\hat{y}) = w(x^*_{\mathsf{NE}})$ and $p(\hat{y}) \le h(x^*_{\mathsf{NE}})$. Since the minimum cost of any solutions for PCST is at most $w(\hat{y}) + p(\hat{y})$, we have $p(y) + w(y) \le \rho'(h(x^*_{\mathsf{NE}}) + w(x^*_{\mathsf{NE}}))$. We can also verify that $h(x_{\mathsf{NA}}) \le p(y)$ holds as in the proof of Theorem 3. For proving $w(x_{\mathsf{NA}}) \le \alpha_{\mathsf{NE}} w(y)$, it suffices to show

$$\left\lfloor \frac{T}{f_i} \right\rfloor \le \alpha_{\mathsf{NE}} \left(\left\lfloor \frac{T}{f_i} \right\rfloor - \left\lfloor \frac{T}{f_{i+1}} \right\rfloor \right). \tag{7}$$

Notice that

$$\frac{1}{q_{i+1} - 1} \cdot \left\lfloor \frac{T}{f_i} \right\rfloor = \left(\frac{q_{i+1}}{q_{i+1} - 1} \right) \cdot \frac{1}{q_{i+1}} \cdot \left\lfloor \frac{T}{f_i} \right\rfloor \ge \frac{q_{i+1}}{q_{i+1} - 1} \cdot \left\lfloor \frac{T}{q_{i+1} f_i} \right\rfloor = \frac{q_{i+1}}{q_{i+1} - 1} \cdot \left\lfloor \frac{T}{f_{i+1}} \right\rfloor.$$

This inequality is equivalent to

$$\left\lfloor \frac{T}{f_{i+1}} \right\rfloor \le \frac{1}{q_{i+1} - 1} \cdot \left(\left\lfloor \frac{T}{f_i} \right\rfloor - \left\lfloor \frac{T}{f_{i+1}} \right\rfloor \right),$$

and the definition of α_{NE} gives

$$\frac{1}{q_{i+1} - 1} \cdot \left(\left\lfloor \frac{T}{f_i} \right\rfloor - \left\lfloor \frac{T}{f_{i+1}} \right\rfloor \right) \le (\alpha_{\mathsf{NE}} - 1) \left(\left\lfloor \frac{T}{f_i} \right\rfloor - \left\lfloor \frac{T}{f_{i+1}} \right\rfloor \right).$$

Combining these inequalities gives the required one. ◀

6 Reducing Capacitated IRP to its Uncapacitated Version

In this section, we consider capacitated IRP. Because of the capacity constraints, we may need more than one tree for visiting nodes in a single round. We assume that these trees used in the same round can share an edge while we need to pay its weight multiple times. When a client node is connected to the root by more than one tree, we have to specify which tree takes care of the demand of this node. In other words, a schedule for a single round consists of a set of trees and an allocation of each client node to one of these trees. The capacity constraints require that the total demand of nodes assigned to a tree does not exceed a given capacity C.

Our main result in this section is a reduction of the capacitated problem to the corresponding uncapacitated version with a slight worsening in the performance ratio. Recall that the vehicle routing cost component of the IRP we model is the minimum-cost Steiner tree rather than the tour. The reduction below applies to other IRPs where the routing is via a tour on the client nodes, but with slightly different factors.

First we present a lower-bound on the tree costs of feasible solutions.

▶ **Lemma 7.** *Let x_C denote any capacitated IRP solution with vehicle capacities C, and $w(r,v)$ denote the weight of the direct edge between r and v in the given metric. Then $w(x_C) \geq \sum_v w(r,v) \sum_{t=0}^{T} d_v^t / C$ (recall that d_v^t is the demand of client v in round t).*

Proof. Consider how every unit of demand to any client is delivered from r in x_C. We "charge" the tree path in the solution from the root to the client scaled by $1/C$ to that unit of demand. Since there are at most C units of demands in any tree, no edge of x gets charged more than once and the paths charged between client i and r have weights at least $w(r,i)$ by the metric property. ◀

We also need the following lemma on partitioning a tree.

▶ **Lemma 8.** *Let U be a rooted tree, and let S denote a set of nodes spanned by U. Suppose that each $v \in S$ has an integer D_v such that $0 \leq D_v \leq C$. Then U can be partitioned into edge-disjoint subtrees $U_1, U_2, \ldots, U_\ell$, and each $v \in S$ can be allocated to one of the subtrees so that*

(i) *$v \in S$ is allocated to the subtree that spans it,*

(ii) *$\sum_{v \in S(U_j)} D_v \leq C$ for each $j = 1, 2, \ldots, \ell$ where $S(U_j)$ denotes the set of nodes in S allocated to U_j,*

(iii) *$\sum_{v \in S(U_j)} D_v \geq C/2$ holds if U_j does not span the root.*

Proof. We prove the lemma by the induction on $|S|$. For $v \in S$, let S_v denote the set of descendants of v in S, and U_v denote the subtree of U which is induced by v and its descendents. Let v^* be a node farthest from the root such that $D_{v^*} + \sum_{v \in S_{v^*}} D_v > C$. If there exists no such v^* (including the case of $|S| = 0$), then we are done.

Suppose not. Then $D_{v^*} > C/2$ or $\sum_{v \in S_{v^*}} D_v > C/2$ holds. Notice that $\sum_{v \in S_{v^*}} D_v \leq C$ holds by the definition of v^*, and $D_{v^*} \leq C$ by the assumption. When the former condition holds, we define the subtree that consists of only v^*, and allocate v^* to this subtree. We then remove v^* from S and apply the induction. When the latter condition holds, we let U_{v^*} be one of the subtrees, and allocate nodes in S_{v^*} to U_{v^*}. We then remove the edges in U_{v^*} from U, and apply the induction. ◀

▶ **Theorem 9.** *Given a γ-approximation for the uncapacitated version of the minimum cost periodic IRP, there is a $(\gamma + 2)$-approximation for the corresponding capacitated version where every tree supports demand at most the given capacity.*

We emphasize that the above reduction applies to all periodic policies, in particular to nested periodic policies. The proof proceeds using a natural combination of the uncapacitated solution along with shortcuts to replenish the supply whenever the vehicle routing solution runs out due to its capacity constraint [1, 24].

Proof. First note that an optimal solution to the uncapacitated counterpart of the given capacitated IRP provides a lower bound on the optimal value of the original capacitated version as well. We first apply the given γ-approximation ignoring the capacities to get an uncapacitated periodic solution to the IRP. Note that this solution defines, without loss of generality, a single tree U that connects the root with all the clients that must be visited in this round.

Let v be a client visited by U in the uncapacitated periodic solution. We assume that a vehicle does not have to deliver more than C units to v in this single visit[1]. We can assume without loss of generality that the uncapacitated solution has this property because we can set $h^v_{st} = +\infty$ when we apply the γ-approximation algorithm if v demands more than C units in rounds from s to t. Since any feasible solutions for the capacitated instance also has the property, this transformation of h^v_{st} makes no effect on the above claim that the uncapacitated solution provides a lower-bound on the optimal value. We define D_v as the units of demands delivered to v by U in the uncapacitated solution. The above assumption implies that $D_v \leq C$.

To complete the algorithm, we need to break every tree U in the uncapacitated solution for the periodic IRP into trees of capacity at most C each. To do this, we employ Lemma 8. Then U is broken into subtrees $U_1, U_2, \ldots, U_\ell$, and each client is allocated to one of the subtrees. For a subtree U_j, let $S(U_j)$ be the set of clients allocated to U_j. If U_j does not span the root, we add the cheapest edge ru_j from a node $u_j \in S(U_j)$ as the "connector" edge to the root to build our capacitated trees of capacity at most C.

Since the subtrees are edge-disjoint, it suffices to show that the weights of the connector edges for all the subtrees can be bounded by twice the lower-bound given in Lemma 7 to get the final guarantee of $\gamma + 2$. For this, observe that we have

$$\sum_{v \in S(U_j)} D_v w(r, v) \geq w(r, u_j) \sum_{v \in S(U_j)} D_v \geq w(r, u_j) \cdot \frac{C}{2},$$

where the last inequality is due to the condition (iii) in Lemma 8. The above inequality simplifies to $w(r, u_j) \leq 2 \sum_{v \in S(U_j)} D_v w(r, v)/C$. A unit of demand is not assigned to more than one subtree simultaneously. This means that the total weight of connector edges is at most twice the lower-bound in Lemma 7. ◀

In order to approximate partition policies, we have to assume that each client v has a constant demand rate d_v per round for a technical reason. Note that several papers [8, 9] assume a constant demand rate per round for the IRP under which all solutions of the same frequency route the same amount of demand in every visit to the same node.

▶ **Theorem 10.** *Given a γ-approximation for the uncapacitated version of the minimum cost partition IRP, there is a $(\gamma + 4)$-approximation for the corresponding capacitated version with constant demand rates where every tree supports demand at most the given capacity.*

---

[1] If we are allowed to split the delivery of the demands to a single node by multiple visits of different capacitated vehicles in the same round, our method can be modified to handle this case with an even better guarantee; we omit discussion of this easier case.

Proof. In Theorem 9, the partitions of trees are possibly different even if the trees visit the same set of clients because the demands of a client can change in different rounds that we employ the tree. Such different partitions are disallowed as solutions to partition policies. In the current setting, this does not happen because a client v of frequency f_i always demands $f_i d_v$ units unless it is at a round in $[T - f_i + 1, T]$. Hence we take the following approach.

When we partition a tree U of frequency f_i, we apply Lemma 8 with $D_v = f_i d_v$ for each client v even if U is used at a round in $[T - f_i + 1, T]$. This way, we always have the same partition for trees of the same frequency in different rounds. Call a tree U the *last tree* if it is used at a round in $[T - f_i + 1, T]$. It can be proven as before that the total weights of connector edges used for augmenting subtrees constructed from trees that are not last trees can be bounded by twice the lower-bound in Lemma 7. For bounding the weights of connector edges used for augmenting subtrees constructed from the last trees, we re-charge units demanded in all rounds. This are at least $D_v = f_i d_v$ units of demands for each client v because $f_i \leq T$. Since a client is not contained by more than one last tree, we do not overuse the demands more than once. Hence the weights of the connector edges for the last trees is also at most twice the lower-bound in Lemma 7. In total, four times the lower-bound is enough for paying the weights of connector edges. ◀

7 Conclusion

We presented constant factor approximation algorithms for finding minimum cost periodic schedules in IRP. A natural question is whether efficient algorithms exist for finding non-periodic schedules. More formally, the problem with non-periodic schedules is defined as follows. For every period in the horizon, we can design a separate tree or tour routes, and the demand for any client at any time is delivered in the last visit before that time to the client in the set of routes. This is an interesting extension of the classic Steiner tree problem and TSP to the round model. It is not difficult to design an $O(\log |V|)$-approximation algorithm for this problem by reducing to the instances with tree metrics using the metric embedding technique [5]. It is an attractive open question to ask if this problem admits a constant factor approximation algorithm. We hope that our ideas presented in the current paper are useful for obtaining an answer to this question.

Acknowledgements. The first author is supported in part by Japan Society for the Promotion of Science (JSPS), Grants-in-Aid for Young Scientists (B) 25730008. The second author is supported in part by grant FA9550-12-1-0411 from the U.S. Air Force Office of Scientific Research (AFOSR) and the Defense Advanced Research Projects Agency (DARPA). The third author is supported in part by NSF grant CCF-1218382.

References

1. K. Altinkemer, B. Gavish. Heuristics with constant error guarantees for the design of tree networks. Management Science 34 (1988) 331–341.
2. S. Anily, A. Federgruen. One warehouse multiple retailer systems with vehicle routing costs. Management Science 36 (1990) 92–114.
3. S. Anily, A. Federgruen. Two-echelon distribution systems with vehicle routing costs and central inventories. Oper. Res. 41 (1993) 37–47.
4. A. Archer, M.H. Bateni, M.T. Hajiaghayi, H. Karloff. Improved approximation algorithms for prize-collecting Steiner tree and TSP, SIAM J. Comput. 40 (2011) 309–332.

5 J. Fakcharoenphol, S. Rao, K. Talwar. A tight bound on approximating arbitrary metrics by tree metrics. J. Comput. Syst. Sci. 69 (2004) 485–497.

6 E. Balas. The prize collecting traveling salesman problem. Networks 19 (1989) 621–636.

7 W.J. Bell, L.M. Dalberto, M.L. Fisher, A.J. Greenfield, R. Jaikumar, P. Kedia, R.G. Mack, P.J. Prutzman. Improving the distribution of industrial gases with an on-line computerized routing and scheduling optimizer. Interfaces 13 (1983) 4–23.

8 L. Bertazzi, M.G. Speranza. Continuous and discrete shipping strategies for the single link problem. Transportation Science 36 (2002) 314–325.

9 L. Bertazzi. Analysis of direct shipping policies in an inventory-routing problem with discrete shipping times. Management Science 54 (2008) 748–762.

10 D. Bienstock, M.X. Goemans, D. Simchi-Levi, D.P. Williamson. A note on the prize collecting traveling salesman problem. Math. Prog. 59 (1993) 413–420.

11 J. Bramel and D. Simchi-Levi. A location based heuristic for general routing problems. Oper. Res. 43 (1997) 649–660.

12 N. Buchbinder, T. Kimbrel, R. Levi, K. Makarychev, M. Sviridenko. Online make-to-order joint replenishment model: primal dual competitive algorithms. SODA 2008, 952–961.

13 L.D. Burns, R.W. Hall, D.E. Blumenfeld, C.F. Daganzo. Distribution strategies that minimize transportation and inventory costs. Oper. Res. 33 (1985) 469–490.

14 A. Campbell, M. Savelsbergh. A decomposition approach for the inventory routing problem. Transportation Science 38 (2004) 488–502.

15 L.M.A. Chan, A. Federgruen, D. Simchi-Levi. Probabilistic analyses and practical algorithms for inventory-routing models. Oper. Res. 46 (1998) 96–106.

16 M. Charikar, S. Khuller, B. Raghavachari. Algorithms for capacitated vehicle routing. SIAM J. Comput. 31 (2001) 665–682.

17 T. Chien, A. Balakrishnan, R. Wong. An integrated inventory allocation and vehicle routing problem. Transportation Science 23 (1989) 67–76.

18 L.C. Coelho, J.-F. Cordeau, G. Laporte, Thirty years of inventory-routing, Transportation Science 48 (2014) 1–9.

19 M. Dror, M. Ball, B. Golden. Computational comparison of algorithms for the inventory routing problem. Annals of Operations Research 4 (1985) 3–23.

20 A. Federgruen, D. Simchi-Levi. Analytical analysis of vehicle routing and inventory management problems. M. O. Ball, T. L. Magnanti, C. L. Monma, G. L. Nemhauser, eds. Network Routing. Handbooks in OR and MS, Vol. 8. North-Holland, Amsterdam, The Netherlands, 297–373 (1995).

21 A. Federgruen, P. Zipkin. A combined vehicle routing and inventory allocation problem. Oper. Res. 32 (1984) 1019–1037.

22 M.X. Goemans, D.P. Williamson, A general approximation technique for constrained forest problems, SIAM J. Comput. 24 (1995) 296–317.

23 M.T. Hajiaghayi, K. Jain. The prize-collecting generalized steiner tree problem via a new approach of primal-dual schema. SODA 2006, 631–640.

24 R. Hassin, R. Ravi, F. S. Salman. Approximation algorithms for a capacitated network design problem, Algorithmica 38 (2004) 417–431.

25 R. Levi, R.O. Round, D.B. Shmoys. Primal-dual algorithms for deterministic inventory problems. Mathematics of Operations Research 31 (2006) 267–284.

26 R. Levi, R. Roundy, D. Shmoys, M. Sviridenko. First constant approximation algorithm for the one-warehouse multi-retailer problem. Management Science 54 (2008) 763–776.

27 R. Roundy. 98%-effective integer-ratio lot-sizing for one-warehouse, multi-retailer systems. Management Science 31 (1985) 1416–1430.

28 A.M. Sarmiento, R. Nagi. A review of integrated analysis of production distribution systems. IIE Transactions 31 (1999) 1061–1074.

29 P. Toth, D. Vigo. The Vehicle Routing Problem. SIAM Monographs on Discrete Mathematics and Applications, Philadelphia, PA (2002).

30 Vender Managed Inventory. http://www.vendormanagedinventory.com/

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

32 D.P. Williamson, D.B. Shmoys. The Design of Approximation Algorithms. Cambridge University Press (2011).