TRANSPORTATION SCIENCE
Vol. 39, No. 2, May 2005, pp. 233–248
ISSN 0041-1655 | EISSN 1526-5447 | 05 | 3902 | 0233
DOI 10.1287/trsc.1030.0080

Shipping Multiple Items by Capacitated Vehicles: An Optimal Dynamic Programming Approach

Shoshana Anily
Faculty of Management, Tel Aviv University, Tel Aviv, Israel, anily@post.tau.ac.il

Michal Tzur
Department of Industrial Engineering, Tel Aviv University, Tel Aviv, Israel, tzur@eng.tau.ac.il*

We consider a system in which multiple items are transferred from a warehouse or a plant to a retailer through identical capacitated vehicles, or by identical freight wagons. Any mixture of the items may be loaded onto a vehicle. The retailer is facing dynamic deterministic demand for several items, over a finite planning horizon. A vehicle incurs a fixed cost for each trip made from the warehouse to the retailer. In addition, there exist item-dependent variable shipping costs and inventory holding costs at the retailer, which are both constant over time. The objective is to find a shipment schedule that minimizes the total cost, while satisfying demand on time.

We address and partially resolve the question regarding the problem's complexity by introducing a dynamic programming algorithm whose complexity is polynomial for a fixed number of items, but exponential otherwise. Our dynamic programming formulation is based on properties satisfied by the optimal solution, and uses an innovative way for partitioning the problem into subproblems.

Key words: logistics; inventory/production; multiple item; dynamic programming
History: Received: July 2002; revision received: March 2003; accepted: June 2003.

Introduction and Literature Review

Consider a system in which items of several types are transferred from a warehouse or a plant to a retailer through identical capacitated vehicles, or by identical freight wagons, each with a finite capacity. We assume that the transportation activity is outsourced to exogenous freight carriers so that the number of vehicles available in a period is unlimited. Any mixture of items may be loaded onto a vehicle as long as the capacity restriction is not violated. The retailer is facing dynamic deterministic demands for several items, over a finite planning horizon. The carrier charges a fixed cost for each vehicle that is dispatched from the warehouse to the retailer, in addition to item-dependent variable shipping costs. Items that are stored at the retailer at the end of a period incur item-dependent inventory holding costs. Both the item-specific shipping costs and inventory holding costs are constant over time. The objective is to find a shipment schedule that minimizes the total cost, while satisfying demand on time.

The same problem may arise in a production environment, in which production decisions of multiple items have to be made, using a certain resource. A fixed cost is associated with the production of a batch (or a partial batch) of items, where each batch is of limited quantity. A specific variant of either the transportation or the production setting that we refer to in the sequel occurs when there is a capacity restriction in each of the periods, i.e., the number of available vehicles or batches in each period is limited to one.

In the case where multiple vehicles may be used in each period (each vehicle has limited capacity, and each incurs a fixed cost), we refer to the problem as the multiple items, with multiple vehicles (MIMV) problem, which is the focus of this paper. The related problem, in which only one vehicle may be used, is referred to as the multiple items with a single vehicle (MISV) problem.

Capacity restrictions exist in most realistic transportation or production systems, but are often ignored. One apparent reason for this phenomenon is the difficulty associated with the consideration of capacitated resources, together with nonstationary demand, even in relatively simple systems. (See the literature review below.) Another issue, which is not well addressed in the literature, is the consideration of transportation in multiple batches, in particular when multiple items are involved. In this paper, we analyze the transportation problem of several different items by multiple capacitated vehicles.

*Currently visiting the Transportation Center and IE/MS Department, Northwestern University.

For the MIMV problem, we identify structural properties that are satisfied by an optimal solution. Those enhance our understanding of the solution arrangement, and lead to algorithmic improvements in our suggested solution procedure. We show a close connection between our problem and the MISV problem, and explain how our results can be used to solve the latter. The importance of our results, in addition to our algorithmic enhancements, are in the insight that we gain into the solution structure, which may be useful for more complex systems with similar characteristics.

The importance of the MIMV problem has increased in recent years, with the growing practice of many manufacturing companies to outsource transportation activities to exogenous freight carriers. In fact, some companies are outsourcing their entire supply-chain activities, in which case the MIMV problem may arise as a subproblem: when the entire system includes multiechelons, for example. The MIMV problem may also be associated with a production facility and a warehouse, where the former is located in a distant location, and the latter is in close proximity to the customers. The problem of shipping different items by capacitated vehicles—possibly via an exogenous logistic company, from the production facility to the warehouse, where they incur holding costs—is well represented by the MIMV problem.

We conclude this section with a literature review on capacitated transportation or production problems, with and without batching considerations. Most of these problems are from the literature on dynamic lot-sizing problems, where production batches stand for vehicles. The single-item with a single batch problem is known in the literature as the capacitated dynamic lot-sizing (CDLS) problem. Two versions of the problem are associated with the cases where the capacities are constant (static) or time varying (dynamic). When T denotes the number of periods in the problem, Florian and Klein (1971) introduced an $O(T^4)$ algorithm for the CDLS with constant capacities, concave production costs, and linear holding costs. Van Hoesel and Wagelmans (1996) improved this result by introducing an $O(T^3)$ algorithm for the same problem with time-varying linear cost parameters. Florian et al. (1980) proved that the same problem under time-varying capacity restrictions is NP-hard. They suggested a nonpolynomial procedure to solve the problem. Chen et al. (1994) suggested a more efficient, though still exponential, dynamic programming approach to the latter problem.

When considering the single item with multiple batches (SIMB) problem, only the static capacity case has been considered in the literature. Pochet and Wolsey (1993) investigated the SIMB problem with time-varying setup, inventory holding, and variable production costs. The problem was shown to be polynomially solvable by finding a shortest path in an appropriately defined network, resulting in an algorithm whose complexity is $O(T^2 \min\{C, T\}) = O(T^3)$, where C is the capacity of a batch. They also showed that the problem can be formulated as a linear programming problem with $O(T^3)$ variables and constraints. Lee (1989) addressed the SIMB problem in which there exists a setup cost for ordering in a particular period, in addition to a different setup cost incurred for each batch; he presented an $O(T^4)$ procedure for the problem.

Proceeding to problems that consider multiple items, the literature becomes very sparse. Constantino (1998) studied the MISV problem in a production setting, where in addition to the total capacity in a period each item that is produced in a given period is subject to lower-bound and upper-bound constraints on its production quantity. Federgruen et al. (2002) studied the MISV problem, where the capacity limits and the cost parameters vary over time. They developed and analyzed a heuristic, which under mild parameter conditions can be designed to be ε-optimal for any $\varepsilon > 0$, with a running time that is polynomially bounded in the size of the problem. Pryor et al. (2000) considered the same MIMV problem as we consider here, suggesting a heuristic as well as an optimal procedure. Both solution methods are based on finding the shortest path in a network with nodes that correspond to periods and arcs that correspond to a schedule from the lower to the higher indexed node. This general shortest path approach is common to many algorithms for this type of problems, including our approach. The difficult part is in computing the arc costs, however (see §2). Their heuristic generalized a heuristic algorithm due to Lippman (1969) for the single item case; it was based on computing heuristically the arc costs in the above network. They also presented a generalization of a search algorithm, which determines the optimal solution for the problem. They did not specify the complexity of the optimal search algorithm. In an accompanying paper to this one, Anily and Tzur (2004) investigated the performance of several algorithms for the MIMV problem, including a new exponential search method.

Yano and Newman (2001) analyzed a more general problem than the MIMV problem, in which the demand for the items dynamically becomes available. However, their algorithm does not necessarily generate the optimal solution for the special case of our problem, as we demonstrate in §2.

Unfortunately, it is still unknown whether even the simplest multiple item versions, i.e., the MISV or the MIMV problems with constant capacity and cost parameters, are polynomially solvable or NP-hard. In this paper we partially resolve this question by introducing for the MIMV problem a dynamic programming formulation whose solution complexity is polynomial for a fixed number of items but exponential otherwise. This means that for a fixed number of items, the problem is polynomially solvable. When the number of items is part of the input, the complexity issue is still unresolved, though our conjecture is that it is NP-hard. In addition, we prove that the MIMV problem is at least as hard as the MISV problem, both under constant capacity and cost parameters. Therefore, the results obtained in this paper for the MIMV problem are also applicable for the MISV problem.

The rest of the paper is organized as follows: In the next section we introduce the problem's notation, an integer linear programming (ILP) formulation of the problem, and some preliminary results on the structure of an optimal policy. In addition, we prove that the problem is at least as hard as the MISV problem with constant capacity. In §2, we propose an algorithm that computes an optimal policy by solving a shortest-path problem on an appropriately defined network. However, computing the arcs' costs of this network is the intricate part of the solution. To compute the arcs' costs we develop, in §3, an exact dynamic programming formulation whose complexity is polynomial for a fixed number of items, but exponential otherwise. Section 4 concludes the paper.

1. Notation and Preliminaries

The MIMV problem is specified by the following parameters:

M = number of items.
T = number of periods in the planning horizon.
d_{it} = demand for item i in period t. We assume that all demands are integers.
p_i = cost of shipping a unit of item i.
h_i = cost of holding in inventory at the retailer a unit of item i at the end of each period.
K = setup cost of dispatching a vehicle (or part of it).
C = capacity of a vehicle, i.e., the number of units that may be loaded in one vehicle.

We assume that all items have the same weight and volume specifications; therefore, a vehicle may contain any mix of items as long as the total number of units does not exceed C. No limitations exist on the number of vehicles that can be dispatched in one period. All demands must be met on time without backlogging. Demand in a given period may be satisfied through shipment in the same period, or from the period's initial inventory. The problem is to find an optimal shipping policy that minimizes the sum of dispatching, variable shipping, and holding costs.

The following decision variables will be used in the ILP formulation below:

$X_i(t)$ = shipping quantity of item i in period t, $1 \le i \le M$ and $1 \le t \le T$;
$Y(t)$ = number of vehicles dispatched in period t, $1 \le t \le T$;
$I_i(t)$ = inventory of item i at the end of period t, $1 \le i \le M$ and $0 \le t \le T$.

We assume that the initial inventory of each item and its final inventory at the end of the horizon are zero, i.e., $I_i(0) = 0$ and $I_i(T) = 0$ for $1 \le i \le M$.

Let $V(X)$ denote the total cost incurred by the shipping quantity matrix X, i.e.,

$$V(X) = \sum_{t=1}^{T}\left(\sum_{i=1}^{M}(p_i X_i(t) + h_i I_i(t)) + KY(t)\right).$$

In the sequel we also use the following notation:

$X(t)$ = total shipping quantity of all items in period t, i.e., $X(t) = \sum_{i=1}^{M} X_i(t)$, $1 \le t \le T$;
$I(t)$ = total inventory of all items at the end of period t, i.e., $I(t) = \sum_{i=1}^{M} I_i(t)$, $1 \le t \le T$.

We also use the following auxiliary notation.

Let $D_i(t', t'') = \sum_{\tau=t'}^{t''-1} d_{i\tau}$ and $D(t', t'') = \sum_{i=1}^{M} D_i(t', t'')$ be the total demand in periods $t', \ldots, t''-1$. We define $D_i(t', t') = 0$.

The MIMV problem can therefore be written as follows:

$$\begin{aligned}
\text{Min} \quad & V(X) = \text{Min} \sum_{t=1}^{T}\left(\sum_{i=1}^{M}(p_i X_i(t) + h_i I_i(t)) + KY(t)\right) && \\
\text{s.t.} \quad & I_i(t) = I_i(t-1) + X_i(t) - d_{it} && 1 \le t \le T,\ 1 \le i \le M \\
& X(t) = \sum_{i=1}^{M} X_i(t) && 1 \le t \le T \\
& X(t) \le CY(t) && 1 \le t \le T \\
& I_i(0) = 0,\ I_i(T) = 0 && 1 \le i \le M \\
& I_i(t) \ge 0,\ X_i(t) \ge 0 && 1 \le t \le T,\ 1 \le i \le M \\
& Y(t) \quad \text{integer} && 1 \le t \le T.
\end{aligned}$$

Due to the fact that the variable shipping costs are static and

$$\sum_{t=1}^{T} X_i(T) = D_i(1, T+1) \quad \forall i,$$

the total variable shipping cost of a feasible solution is constant, namely,

$$\sum_{i=1}^{M} p_i D_i(1, T+1).$$

Therefore, we ignore this cost component in our future analysis. In other words, the only costs that affect the cost of a solution are the vehicle-dispatching costs and the items' holding costs. (In the sequel, we use the terms solution, policy, and schedule interchangeably; they have the same meaning.) A policy can be described by a matrix X of M rows and T columns where the (i,t)th element in the matrix is $X_i(t)$.

The above ILP formulation for MIMV contains as many general integer decision variables as the number of periods in the problem (the $Y(t)$ variables). Therefore, the solution time, when solved through general ILP software, may be relatively large. In a preliminary study that we performed, we ran the above ILP using the AMPL software with the CPLEX solver. For problems with number of periods and number of items ranging from 10 to 50, we observed that running times were variable, and ranged from 1 second to over 5 hours. Moreover, even for problem instances with the same parameter characteristics, there was a large variability in solution time. Our conclusion is that although some problems may be solved through ILP, this is not an appropriate solution method for this problem in general.

We use the term *full vehicle* to refer to a vehicle loaded by a quantity of C units, and the term *partial vehicle* to refer to a vehicle that is loaded by a positive quantity of less than C units. Therefore, a shipment of $X(t)$ units in any period t consists of $\lfloor X(t)/C \rfloor$ full vehicles. If $X(t) \bmod C > 0$, then the shipment consists, in addition, of a partial vehicle loaded by $X(t) \bmod C$ units.

Assume that the items are numbered in a nondescending order of their holding cost rates. Note that different items with identical holding cost rates can be combined into a single item, since they are identical for all computational purposes. Therefore, we assume without loss of generality that $0 < h_1 < h_2 < \cdots < h_M$.

We are now ready to present our first lemma, which consists of four important properties—denoted as (P1)–(P4)—that are satisfied by an optimal solution. The first property is an extension by Pryor et al. (2000) to a similar property that was proved by Baker et al. (1978) for the CDLS problem. Property (P4) was proved in Pryor et al. (2000).

Lemma 1. *For any optimal policy X and associated optimal inventory matrix I, the following properties hold*:

Property (P1).

$$I(t-1)(X(t) \bmod C) = 0 \quad 1 \le t \le T$$

Property (P2).

$$X(t) - \sum_i (d_{it} - I_i(t-1))^+ < \begin{cases} X(t) \bmod C & \text{if } X(t) \bmod C > 0 \\ C & \text{if } X(t) \bmod C = 0 \end{cases} \quad 1 \le t \le T$$

Property (P3).

$$X(t) \le \lceil D(t, t+1)/C \rceil C \quad 1 \le t \le T$$

Property (P4).

$$I_M(t) < C \quad 1 \le t \le T.$$

Proof. Property (P1) states that if the initial inventory in period t is positive, then the shipping quantity in period t consists of full vehicles only; otherwise, we could reduce the total costs by delaying the shipment of some of the items that are carried from the previous period. For the same reason, if the optimal policy is to ship a partial vehicle then the initial inventory in that period should be zero. Property (P2) states that an optimal schedule never dispatches a vehicle whose entire content is for future periods. This is true since otherwise we could reduce the total costs by postponing the shipment of the entire vehicle. Property (P3) states that the maximum shipment quantity never exceeds the total capacity of the minimum number of full vehicles required to cover the demand in that period. Otherwise, we could reduce the total costs by delaying the shipment of at least one vehicle. This is in fact a special case of (P2), when the starting inventory in a period is zero. Finally, Property (P4) states that the amount that we hold in inventory from the most expensive item is always less than the vehicle capacity. □

Note that as a result of the above lemma the ending inventory of an optimal schedule in two consecutive periods t and $t+1$ must satisfy $I(t+1) < I(t) + C$.

In the next lemma we show that, given the total shipping quantities $X(t)$ for each period t, $1 \le t \le T$, it is beneficial to delay the shipment of *expensive* items (i.e., items with high inventory holding cost rates) as much as possible. A similar property was presented in Pryor et al. (2000).

Lemma 2. *Consider two feasible policies X and X' such that $X_j(t') > 0$ and $X_i(t'') > 0$ for $1 \le t' < t'' \le T$ and $h_i < h_j$ for two items $1 \le i < j \le M$. If policy X' is identical to policy X except for the following elements, $X'_j(t') = X_j(t') - 1$; $X'_i(t') = X_i(t') + 1$; $X'_j(t'') = X_j(t'') + 1$; $X'_i(t'') = X_i(t'') - 1$, then $V(X') < V(X)$.*

Proof. The changes in policy X' (compared to policy X) are: (i) One unit of item i is shipped in period t' instead of in period t'' and (ii) one unit of item j is shipped in period t'' instead of in period t'. Since both policies are feasible, the first change incurs an additional holding cost of h_i for each of the periods t', $t'+1$, ..., $t''-1$, and the second change saves a holding cost of h_j for each of the periods t', $t'+1$, ..., $t''-1$. Therefore, $V(X') - V(X) = (t'' - t')(h_i - h_j) < 0$. □

Corollary 1. *For any optimal policy X and associated optimal inventory matrix I, the following holds*: *If for some period t we have that $X_i(t) > 0$ for some item i, then for items $j > i$, $I_j(t-1) = 0$.*

Based on Lemma 2, and given a vector $(X(1), \ldots, X(T))$ of aggregated quantities shipped in each period, the Scheduling algorithm described below computes the best detailed schedule for each period, that is, it determines the shipment quantities of each item in each period. Given a vector $(X(1), \ldots, X(T))$, the algorithm uses the following procedure to determine the quantities $X_i(t)$ for $1 \leq i \leq M$ and $1 \leq t \leq T$: Start at the last period and allocate to it $X(T)$ units by giving priority to the most expensive items that are demanded in this period (i.e., start with the most expensive item and allocate to it its demand, continue with the next expensive item, and so on, down to the least expensive item). If $X(T) < D(T, T+1)$, then shift the excessive demand in period T, i.e., $D(T, T+1) - X(T)$ least expensive units, to period $T-1$ and repeat the process backward to period 1. A formal description of the algorithm is given in Appendix A.

The complexity of the scheduling algorithm is $O(TM)$, given that the items are ordered in a nondescending order of their inventory holding costs. Otherwise, it is $O(MT + M \log M)$.

We conclude this section by proving that under static capacity and cost parameters the MIMV problem is at least as hard as the MISV problem, i.e., any algorithm that solves the MIMV problem is also capable of solving the MISV problem. Note first that the MISV problem is feasible if, and only if, for any t, $1 \leq t \leq T$, $D(1, t+1) \leq tC$, i.e., the total demand over the first t periods does not exceed the cumulative maximum load that is allowed by the vehicle capacity during these periods. Our first step is to prove that it is sufficient to consider only MISV instances for which the demand in each period does not exceed C. To do this, we present an algorithm, referred to as the Shifting algorithm, which is applied to any feasible instance of the MISV problem to get another instance of MISV, in which the demand in each period does not exceed C. This is done recursively by shifting the demand for the least expensive items from a period whose total demand is greater than C to the previous period. The complexity of the algorithm is $O(TM)$. A formal description of the algorithm is given in Appendix B. Lemma 3 below proves that both instances have the same optimal shipment schedules.

Lemma 3. *Suppose we are given a feasible instance* $\Pi = \{D_i(t, t+1)\colon 1 \leq t \leq T,\ 1 \leq i \leq M\}$ *of the MISV problem for which there exists at least one period* t, $2 \leq t \leq T$, *such that* $D(t, t+1) > C$. *By applying the Shifting algorithm we obtain a new instance of the MISV problem* $\Pi' = \{D'_i(t, t+1)\colon 1 \leq t \leq T,\ 1 \leq i \leq M\}$ *with* $D'(t, t+1) \leq C$ *for* $1 \leq t \leq T$. *Then a shipment schedule is optimal for instance* Π' *if and only if it is optimal for instance* Π.

Proof. We first prove that the optimal solution to Π' is also optimal to Π. Our proof is based on the following claims: (i) The cost of any solution X that is feasible for both instances Π and Π' incurs under Π a higher cost than under Π'. However, the cost difference is a constant, which does not depend on X. (ii) Any feasible solution for Π' is also feasible for Π but not vice versa. (iii) The optimal solution for Π is feasible for Π'.

As a result of the above three claims, the set of feasible solutions to instance Π' is a subset of the set of feasible solutions to instance Π, but it contains all optimal solutions of Π. Since the costs of any solution to Π and Π' differ by a constant, the optimal solution for Π' is also optimal for Π.

Claim (i) follows from the fact that a unit that was shifted from, say, period t to period $t' < t$ incurs under Π the additional cost (compared to Π') of being held in inventory for $(t - t')$ periods. The sum of these inventory holding costs over all units that were shifted is the cost difference, which depends only on the identity of the shifted units, their holding cost rates, and the number of periods by which they were shifted, but not on the solution.

Claim (ii) follows since, given a feasible solution to Π', we obtain a feasible solution to Π by just keeping in inventory the units that were shifted. Not every solution to Π is feasible for Π', however, since under Π' a unit that was shifted from t to t' has to be shipped by period t', while it may be shipped after t' (but before t) according to the solution to Π.

To prove claim (iii), note that due to the capacity restrictions, the same number of units that were shifted by the Shifting algorithm have to be shipped early by any feasible solution to Π. Moreover, they have to be shipped as early as the period to which they were shifted. According to Lemma 2 it is preferable to shift first the units of the least expensive items, as is done by the Shifting algorithm.

To prove that the optimal solution to Π is also optimal to Π', we note again that the set of feasible solutions for Π' is a subset of the feasible solutions for Π, that the optimal solution for Π is feasible for Π', and that the cost of each feasible solution for Π' differs from the cost for Π only by a constant. □

The next lemma is based on Lemma 3, and proves the desired relationship between the MIMV and MISV problems.

Lemma 4. *For static capacity and cost parameters, the MIMV problem is at least as hard as the MISV problem. That is, if the MISV problem is NP-hard, then the MIMV problem is also NP-hard.*

Proof. We show that any algorithm that solves the MIMV problem can also be used to solve the MISV problem. As a result, if no polynomial algorithm exists for the MISV problem (unless P = NP), it implies, also, that no polynomial algorithm exists for the MIMV problem because otherwise it could also be applied to the MISV problem. Thus, if the MISV problem is NP-hard, the MIMV problem is also NP-hard.

To show that any algorithm that solves the MIMV problem can also be used to solve the MISV problem, recall that by Lemma 3 and the fact that the Shifting algorithm is polynomial, it is sufficient to consider MISV problems in which the total demand in each period does not exceed the capacity limit. Thus we consider such instances from now on in the proof. By applying the optimal algorithm for MIMV on such an instance of MISV, we get a schedule in which at most one vehicle is dispatched in each period. This is due to the fact that the total demand in each period does not exceed C, and with two dispatched vehicles in a given period all the content of the second vehicle is for future use, contradicting the structure of an optimal schedule (see (P2) of Lemma 1). Because the MISV problem is more restricted than the MIMV problem, but the solution of the latter is feasible for the former, we conclude that it is also optimal to the former. □

As mentioned in the introduction, the complexity of the MISV problem is still unknown. Here we have shown that any method for the MIMV problem can also be applied to the MISV problem. In particular, one can apply the dynamic programming formulation suggested in §3, which is polynomial for a fixed number of items but exponential otherwise.

2. Formulation as a Shortest-Path Problem

In this section we show how to find an optimal policy by solving a shortest-path problem on a network with nodes $1, 2, \ldots, T+1$. In the proposed network there exist arcs connecting pairs of nodes t' and t'' with $1 \le t' < t'' \le T+1$. An arc connecting node t' to node t'' represents the minimum cost schedule between period t' and period $t''-1$, given $I(t'-1) = I(t''-1) = 0$ and $I(\tau) > 0$ for $t'-1 < \tau < t''-1$. The cost on this arc consists of all dispatching and inventory holding costs incurred in periods $t', t'+1, \ldots, t''-1$. An optimal schedule for arc (t', t'') dispatches the minimum possible number of vehicles in periods $t', \ldots, t''-1$ to cover the demand in these periods. This observation follows from Property (P1) of Lemma 1, which ensures that the shipping quantity in periods $t'+1, \ldots, t''-1$ should all be full vehicles (i.e., integer multiples of C). Only in period t' can a partial vehicle be dispatched, since then the starting inventory is zero. Thus, the total setup cost of arc (t', t'') is $K\lceil D(t', t'')/C \rceil$. If the total holding costs on the arcs were given to us, we could obtain the optimal solution by applying a shortest-path algorithm that requires $O(T^2)$ time. (See the example at the end of this section, illustrating the shortest-path construction.)

As mentioned in the introduction, Yano and Newman (2001) solved a more general problem than the MIMV that we consider here. Their solution method, when applied to our MIMV problem, resolves the issue of computing the holding costs on the arcs by employing a simple procedure that dispatches the vehicles as late as possible without causing a backlogging. Indeed, for the SIMB problem (i.e., $M = 1$) it is easy to see that this proposed schedule is optimal for all possible arcs. However, we use the following example by Pryor et al. (2000) to demonstrate that delaying shipments as much as possible is not necessarily optimal for all arcs in the network when multiple items are involved. Consider an arc of four periods and two items with $C = 10$, $h_1 = 1$, $h_2 = 100$, demand for Item 1: 0, 0, 6, 6 in Periods 1–4, respectively, and demand for Item 2: 4, 4, 0, 0 in Periods 1–4, respectively. According to Yano and Newman's (2001) schedule, shipments are delayed as much as possible, thus 10 units are shipped in Periods 1 and 3 and none in Periods 2 and 4. More specifically, eight units of Item 2 and two units of Item 1 are shipped in Period 1, and 10 units of Item 1 in Period 3. However the optimal solution for this problem is to ship 10 units in each of the first two periods (four units of Item 2 and six units of Item 1 in each period).

As discussed below, the intricate part of the solution for this problem is the computation of the holding cost of the arcs. If this part of the algorithm were polynomial the whole procedure would be polynomial. In §3 we propose a dynamic programming formulation for computing the holding cost of an arc, which is polynomial for a given number of items, but exponential otherwise, implying the same complexity statement for the overall procedure. However first we show that some of the arcs are not admissible, and therefore may be removed from the network. This result will help us enhance the computation of the dynamic programming formulation of §3.

Removing Nonadmissible Arcs

Consider an arc (t', t'') in the network. Clearly $X(t') \ge D(t', t'+1)$ because the initial inventory in period t' is 0; that is, all demand in period t' must be covered from a shipment in period t'. The next lemma specifies the exact quantity shipped in period t'.

Lemma 5. *Suppose that in an optimal policy X^* we have two periods t' and t'' ($t' < t''$) such that $I(t'-1) = I(t''-1) = 0$ and $I(\tau) > 0$ for $t'-1 < \tau < t''-1$. Then,*
(a) $X^*(t') \bmod C = D(t', t'') \bmod C$
(b) $D(t', t'+1) \le X^*(t') \le \lceil D(t', t'+1)/C \rceil C$.

Proof. Part (a) follows from the fact that the initial inventory in period t' and the final inventory in period $t''-1$ are both zero, and the shipment quantities in periods $t'+1, \ldots, t''-1$ are all in full vehicles.

The lower bound in Part (b) holds for any feasible solution since $I(t'-1)=0$. For the upper bound, see Property (P3) in Lemma 1. $\square$

The following two corollaries are immediate from Lemma 5:

Corollary 2. *If $D(t',t'') \bmod C < D(t',t'+1) \bmod C$ and $D(t',t'') \bmod C \neq 0$, then arc (t',t'') is not part of the optimal solution.*

Proof. Recall that an arc connecting node t' to node t'' represents the minimum cost schedule between period t' and period $t''-1$ given $I(t'-1)=I(t''-1)=0$ and $I(\tau)>0$ for $t'-1<\tau<t''-1$. If $D(t',t'') \bmod C < D(t',t'+1) \bmod C$ and $D(t',t'') \bmod C \neq 0$, then by Lemma 5(a) and the left inequality of Lemma 5(b), $X^*(t') = \lceil D(t',t'+1)/C \rceil C + D(t',t'') \bmod C$, which contradicts the right inequality of Lemma 5(b). $\square$

Corollary 3. *The only arc emanating from a node t' with $D(t',t'+1) \bmod C = 0$ is the arc $(t',t'+1)$.*

Proof. Consider a period t' such that $D(t',t'+1) \bmod C = 0$ and suppose that there exists an arc (t',t'') for $1 \le t' < t'' \le T+1$ in the network. Then this arc represents a policy for which $I(t'-1)=I(t''-1)=0$ and $I(\tau)>0$ for $t' \le \tau < t''-1$. Suppose by contradiction that $t''>t'+1$. This assumption implies that $I(t')>0$, and therefore, $X(t')>D(t',t'+1)$. However, in view of Lemma 5(b), and because the total demand in period t' is an integer multiple of C, we obtain $X(t')=D(t',t'+1)$, a contradiction. Thus, $t''=t'+1$. $\square$

To illustrate the results of this section, consider a problem that consists of only Items 1 and 2 from the example provided in Appendix C. The network associated with this example consists of Nodes 1, 2, 3, and 4 and Arcs (1,2), (1,3), (1,4), (2,3), (2,4), and (3,4). The total setup cost of these arcs are K, $2K$, $4K$, K, $4K$, and $3K$, respectively. However, note that Arcs (1,3) and (2,4) are not admissible since the whole content of the last (partial) vehicle shipped in its first period is for future use. This demonstrates the use of Lemma 5 and the reduction in the size of the network that can be achieved. In this case, only two paths in the network have to be considered: (1,4) and (1,2)–(2,3)–(3,4). Using the dynamic programming we provide below, we calculated the optimal holding costs for the arcs that need to be considered; they are equal to $5h_1=5$ (Arc (1,4)), and zero for all other arcs. Assuming $K=20$, we get the following total costs (denoted locally by $c(t',t'')$) for the arcs: $c(1,4)=85$, $c(1,2)=20$, $c(2,3)=20$, and $c(3,4)=60$. Now, solving a standard shortest-path problem on this network, we conclude that the optimal path is to use Arc (1,4) with total cost of 85. The specific shipping quantities in each of the periods for each of the items are also found through the solution of the dynamic programming algorithm.

3. An Exact Dynamic Programming Formulation

In this section we present an exact dynamic programming formulation that computes the total holding cost of an arc. The procedure is based on the properties of an optimal solution, developed in the previous sections. For simplicity, we denote in this section a general arc (t',t'') in the network by $(1,T+1)$.

Definition 1. Given a shipment schedule for periods $1,\dots,t$, let $\text{Indx}(t)$ be the most expensive item held in inventory at the end of period t. If the inventory at the end of period t is zero, then $\text{Indx}(t)$ is set to 1. Formally,

$$\text{Indx}(t) = \begin{cases} \max\{i\colon I_i(t)>0\} & \text{if } I(t)>0 \\ 1 & \text{otherwise.} \end{cases}$$

Definition 2. Given a shipment schedule for some initial periods, let τ_j $1 \le j \le M$ be the first period after those initial periods, in which at least one item from the set $\{1,\dots,j\}$ is shipped, and $\tau_0 = T+1$. The sequence τ_j is nonincreasing in j by definition. Initially we set $\tau_j = 1$ for $1 \le j \le M$. If for some item j and some t, $I_m(t-1) = D_m(t,T+1)$ for all $m \le j$, then at the end of period $t-1$ we set $\tau_m = T+1$ for all $m \le j$.

Note that at the end of period $t>1$, some τ values might be updated as a result of decisions made in period t, while the other τ values might remain unchanged.

Definition 3. Given the values $\tau_M \le \tau_{M-1} \le \cdots \le \tau_1$ at the end of period $t-1$, let $\kappa(t) = \min\{i\colon \tau_i = t\}$. That is, $\kappa(t)$ is the least expensive item shipped in period t. (Except maybe initially, when $\tau_j = 1$ for $1 \le j \le M$ by definition. In this case, if the demand of Item 1 in the first periods is zero, we may choose not to ship Item 1 in Period 1.) If $\tau_M > t$, then set $\kappa(t) = M+1$, which means that no shipment occurs in period t. For ease of notation, we use κ instead of $\kappa(t)$ and denote the time index only for periods other than t.

Note that $\kappa = \kappa(t) \ge \text{Indx}(t-1)$, as follows from Corollary 1. Note also that by definition, $\tau_{\kappa-1} > t$.

Lemma 6. *$I_j(t-1)$ satisfies $D_j(t,\tau_j) \le I_j(t-1) \le D_j(t,\tau_{j-1})$ for all $j<\kappa$.*

Proof. The lemma is a result of the definitions of κ and the τ's. $\square$

When developing our dynamic programming formulation, it is clear that at the beginning of period t it is required to know the initial inventories of all items, i.e., the value of $I_m(t-1)$ for all $m=1,\dots,M$. However, proceeding with a straightforward dynamic programming formulation that considers all possible shipment quantities will result in a very inefficient procedure. Thus, in the following definitions, the

τ values defined earlier are used to decompose the problem into smaller subproblems.

Let $F(t, \tau_1, \tau_2, \dots, \tau_M, I_1(t-1), I_2(t-1), \dots, I_M(t-1))$ = the minimum holding cost incurred by items that are shipped in periods $t, \dots, T$, when at the end of period $t-1$ the inventory of item i is $I_i(t-1)$ and the first period in which at least one item from the set $\{1, \dots, i\}$ is shipped is in period τ_i, for all $1 \le i \le M$.

For $\tau_{\kappa-1} > t$, let $G(t, \tau_{\kappa-1}, \kappa, I_\kappa(t-1))$ be the minimum holding cost incurred by items $\kappa, \dots, M$ shipped in periods $t, \dots, \tau_{\kappa-1}-1$, when in period t the initial inventory of item κ is $I_\kappa(t-1) \ge 0$ and the initial inventory of items $\kappa+1, \dots, M$ is zero. Note that these items satisfy the demand of these periods only: that is, no units are shipped in these periods for consumption in periods later than $\tau_{\kappa-1}$, i.e., $I_m(\tau_{\kappa-1}-1) = 0$ for all $m \ge \kappa$. Period $\tau_{\kappa-1}$ is the first period after t where an item in $\{1, \dots, \kappa-1\}$ is shipped. (In such a period the beginning inventory of all items m, $m \ge \kappa$ must be zero; see Corollary 1.) We let $G(t, t, ., .) = 0$; in addition, $G(t, \tau_M, M+1, 0) = 0$ because no shipment occurs in periods $t, \dots, \tau_M - 1$.

Given the above two definitions, Theorem 1 provides us with a way to decompose a certain problem into several smaller subproblems.

Theorem 1.

$$
\begin{aligned}
F(t, \tau_1, \tau_2, \dots, \tau_M, I_1(t-1), I_2(t-1), \dots, I_M(t-1))& \\
= G(t, \tau_{\kappa-1}, \kappa, I_\kappa(t-1))& \\
+ \sum_{j=1}^{\kappa-1} G(\tau_j, \tau_{j-1}, j, I_j(t-1) - D_j(t, \tau_j))&
\end{aligned}
$$

with κ as in Definition 3 above.

Proof. Consider one term from the summation expression associated with periods $\tau_j, \dots, \tau_{j-1}-1$. For this term we show that all restrictions imposed by the $G(\cdot)$ function are a result of properties of the optimal solution.

If $\tau_j = \tau_{j-1}$, then no periods are included in $\tau_j, \dots, \tau_{j-1}-1$, and by our convention $G(t, t, ., .) = 0$. Otherwise, for $\tau_j < \tau_{j-1}$, only items $j, \dots, M$ are shipped in periods $\tau_j, \dots, \tau_{j-1}-1$ since by definition of τ_{j-1}, no shipment of items $1, \dots, j-1$ occurs before period τ_{j-1}. In period τ_j item j is shipped the first time in or after period t, therefore its inventory at the beginning of period τ_j is $I_j(t-1) - D_j(t, \tau_j)$. In addition, the beginning inventory at period τ_j is zero for items that are more expensive than j, as implied by the $G(\cdot)$ expression, and justified by Corollary 1. At the end of period $\tau_{j-1}-1$, the $G(\cdot)$ function requires that $I_m(\tau_{j-1}-1) = 0$ for all $m \ge j$, which is justified by considering the next set of periods, i.e., periods $\tau_{j-1}, \dots, \tau_{j-2}-1$, and repeating the argument with respect to the beginning inventory.

The above justification applies for all j in the summation expression, as well as for the first term, associated with periods $t, \dots, \tau_{\kappa-1}-1$. □

Due to Theorem 1 we are able to consider all $G(\cdot)$ functions separately. Computing the $G(\cdot)$ functions is in fact the core of the problem and most of the work. Solving the entire problem is associated with finding the value of $F(1, 1, \dots, 1, 0, \dots, 0)$. Note that initially, $\tau_j = 1$ for $1 \le j \le M$ and $I_j(0) = 0$ for $1 \le j \le M$, thus $\kappa = \kappa(1) = 1$ and $\tau_{\kappa-1} = \tau_0 = T+1$, so that $F(1, 1, \dots, 1, 0, \dots, 0)$ is equivalent to $G(1, T+1, 1, 0)$. We present next a recursive functional equation to calculate it.

Calculation of $G(t, \tau_{\kappa-1}, \kappa, I_\kappa(t-1))$

Note that when calculating a certain $G(t, \tau_{\kappa-1}, \kappa, .)$ function, all values of τ_i for $i < \kappa$ are already determined from the shipment schedule of periods $1, \dots, t-1$. The decision in period t may only change the τ value of items i with $\tau_i = t$ (i.e., $i \ge \kappa$).

The idea behind the efficient computation of the $G(\cdot)$ functions is that under the optimal policy only certain identity and quantity combinations are possible for items held in inventory. We organize these possible combinations in a list, defined as follows:

Let $L(t, \tau_{\kappa-1}, \kappa, I_\kappa(t-1))$ for $\tau_{\kappa-1} > t+1$, be a list of all possible allocations (into specific items) of inventory at the end of period t of all items m, $m \ge \kappa$ that were shipped in period t for periods $t, \dots, \tau_{\kappa-1}-1$, when in period t the initial inventory of item κ is $I_\kappa(t-1) \ge 0$, and of items $\kappa+1, \dots, M$ is zero. A member in this list specifies the most expensive item held in inventory at the end of period t (i.e., $\mathrm{Indx}(t)$), the inventory levels of all items m, $\kappa \le m \le \mathrm{Indx}(t)$ at the end of period t (i.e., $I_\kappa(t), I_{\kappa+1}(t), \dots, I_{\mathrm{Indx}(t)}(t)$), and the period for each item $i \ge \kappa$, in which it, or a less expensive item, will be shipped next (denoted as defined above by $\tau_\kappa, \tau_{\kappa+1}, \dots, \tau_M$). Note again that the values $\tau_1, \dots, \tau_{\kappa-1}$ that are already known at the end of period $t-1$, are not affected by the shipment in period t because items $1, \dots, \kappa-1$ are not shipped in period t. The newly determined sequence $\tau_\kappa, \tau_{\kappa+1}, \dots, \tau_M$ specifies the value of $\kappa(t+1)$, which is required for the cost calculation in period $t+1$. Thus, a member in this list is referred to as

$$
\begin{aligned}
&(\kappa, \mathrm{Indx}(t), \kappa(t+1), I_\kappa(t), \dots, I_{\mathrm{Indx}(t)}(t), \tau_\kappa, \tau_{\kappa+1}, \dots, \tau_M) \\
&\qquad \in L(t, \tau_{\kappa-1}, \kappa, I_\kappa(t-1)).
\end{aligned}
$$

Note that the τ values in an element of the list $L(t, .)$ are associated with the end of period t. If $\tau_{\kappa-1} = t+1$, we use an initial condition to solve the associated $G(\cdot)$ problem (see below).

The list of all possible allocations is constructed by enumerating all possible values of all the parameters that specify an element in the list, excluding combinations of values that do not satisfy the optimality conditions discussed throughout the paper. Later in this section we show how to find the elements that construct this list, and demonstrate that there are relatively few such elements. For now, we assume that the list of elements exists, and use it in the dynamic programming formulation, presented in Equation (1) below.

For $\tau_{\kappa-1} > t+1$, let $H(t, I_\kappa(t-1), \text{Indx}(t), I_\kappa(t), \ldots, I_{\text{Indx}(t)}(t))$ = the holding costs incurred by units of items κ, $\kappa+1, \ldots, \text{Indx}(t)$ that are shipped in period t, when in period t the initial inventory of item κ is $I_\kappa(t-1) \geq 0$, the initial inventory of items that are more expensive than κ is zero, and the inventory at the end of period t of items $\kappa, \ldots, \text{Indx}(t)$ is $I_\kappa(t), \ldots, I_{\text{Indx(t)}}(t)$, respectively. Note that $\tau_{\kappa-1} > t+1$ implies that $\text{Indx}(t) \geq \kappa$ and therefore the function $H(\cdot)$ is well defined. If $\kappa = M+1$, then set $H(t, \cdot) = 0$ because it means that no items are shipped in period t. Note that the $H(t, \cdot)$ function represents the holding costs incurred by units that are shipped in period t, and until the units are consumed.

Given each item's inventory at the end of period t, it is clear when each unit of it will be consumed (assuming (without loss of generality) a first-in-first-out (FIFO) mechanism). Therefore, the value of $H(t, I_\kappa(t-1), \text{Indx}(t), I_\kappa(t), \ldots, I_{\text{Indx(t)}}(t))$ may be calculated as follows (see the explanation following the expression):

$$\begin{aligned}
&H(t, I_\kappa(t-1), \text{Indx}(t), I_\kappa(t), \ldots, I_{\text{Indx}(t)}(t))\\
&\quad= \sum_{i=\kappa+1}^{\text{Indx}(t)} \sum_{\tau=t}^{T+1} h_i(I_i(t) - D_i(t, \tau))^+\\
&\qquad+ \sum_{\tau=t}^{T+1} h_\kappa[I_\kappa(t) - (I_\kappa(t-1) - D_\kappa(t, \tau))^+\\
&\qquad\qquad - (D_\kappa(t, \tau) - I_\kappa(t-1))^+]^+.
\end{aligned}$$

According to the definition of $H(\cdot)$, the holding costs in the above expression consider only items that were shipped in period t. In the first part we account for the costs of holding items $\kappa+1, \ldots, \text{Indx}(t)$ in periods $t, \ldots, T$. In the second part we account for the costs of holding item κ in periods $t, \ldots, T$, an expression that is calculated as explained next. The term $(I_\kappa(t-1) - D_\kappa(t, \tau))^+$ represents the number of units of item κ that are held at the end of period $\tau-1$ and were shipped before period t, therefore should not be included in the $H(\cdot)$ function. The term $(D_\kappa(t, \tau) - I_\kappa(t-1))^+$ represents the net demand from period t to $\tau-1$, as observed at the end of period $t-1$. At most one of these terms is positive. In either case (whichever is positive), the term $[I_\kappa(t) - (I_\kappa(t-1) - D_\kappa(t, \tau))^+ - (D_\kappa(t, \tau) - I_\kappa(t-1))^+]^+$ represents the number of units of item κ that were shipped in period t and are held in inventory at the end of period $\tau-1$.

We are now ready to present the general functional equation of the dynamic programming formulation for the $G(\cdot)$ function. The equation is given in three parts, (1a)–(1c), referred to together as Equation (1).

$$\begin{aligned}
&G(t, \tau_{\kappa-1}, \kappa, I_\kappa(t-1)) = \infty\\
&\qquad \text{if } t \neq 1 \text{ and } \bigg((D_\kappa(t, \tau_{\kappa-1}) - I_\kappa(t-1))^+\\
&\qquad\qquad + \sum_{i=\kappa+1}^{M} D_i(t, \tau_{\kappa-1})\bigg) \bmod C \neq 0 \qquad (1a)
\end{aligned}$$

$$\begin{aligned}
&G(t, \tau_{\kappa-1}, \kappa, I_\kappa(t-1)) = 0\\
&\qquad \text{if } t \neq 1,\ \tau_{\kappa-1} = t+1 \text{ and}\\
&\qquad\qquad \bigg((d_{kt} - I_\kappa(t-1))^+ + \sum_{i=\kappa+1}^{M} d_{it}\bigg) \bmod C = 0 \qquad (1b)
\end{aligned}$$

otherwise (i.e., $\tau_{\kappa-1} > t+1$, thus $\text{Indx}(t) \geq \kappa$),

$$\begin{aligned}
&G(t, \tau_{\kappa-1}, \kappa, I_\kappa(t-1)) = \operatorname*{Min}_{\substack{(\kappa, \text{Indx}(t), \kappa(t+1), I_\kappa(t), \ldots, I_{\text{Indx}(t)}(t),\\ \tau_\kappa, \tau_{\kappa+1}, \ldots, \tau_M) \in L(t, \tau_{\kappa-1}, \kappa, I_\kappa(t-1))}}\\
&\quad H(t, I_\kappa(t-1), \text{Indx}(t), I_\kappa(t), \ldots, I_{\text{Indx}(t)}(t))\\
&\qquad + G(t+1, \tau_{\kappa(t+1)-1}, \kappa(t+1), I_{\kappa(t+1)}(t))\\
&\qquad + \sum_{i=\kappa}^{\kappa(t+1)-1} G(\tau_i, \tau_{i-1}, i, I_i(t) - D_i(t+1, \tau_i)) \qquad (1c)
\end{aligned}$$

Explanation and Justification

Recall that we use the $G(t, \cdot)$ function for calculating the value of $F(t, \cdot)$, according to Theorem 1. In the proof of the above theorem, it was demonstrated that the restrictions imposed by the definition of the $G(\cdot)$ function are justified by the optimality conditions. It remains to justify, given the restrictions imposed by the definition of the $G(\cdot)$ function, its calculation according to Equation (1). Recall that the total dispatching cost for a given subproblem is constant, and therefore computed separately, as explained in §2. Therefore, the $G(\cdot)$ function includes (also by definition) only holding costs.

The decision at the minimization stage is how much to ship in period t from each of the items $\kappa, \ldots, M$, the only items that may be shipped in that period. Equation (1a) represents the case where there is no feasible schedule that satisfies the optimality condition of dispatching only full vehicles (when $t \neq 1$, i.e., t is not the first period of the subproblem). To exclude any solution that makes use of such a schedule, we assign to it a cost of ∞. Equation (1b) is associated with the case of $\tau_{\kappa-1} = t+1$, in which modulo C of the net demand in period t is zero, therefore the cost is zero. Note that $\tau_{\kappa-1} = t+1$ implies that $\mathrm{Indx}(t) < \kappa$, a case that is not covered by (1c) and therefore is treated separately. In (1c), once a shipment is performed in period t, we may have at the end of period t inventory from all items shipped, and as above we denote the most expensive item for which inventory is held by $\mathrm{Indx}(t)$. For each item m, $\kappa \le m \le \mathrm{Indx}(t)$, its shipment quantity in period t determines the amount of inventory at the end of period t, as well as its τ value. Such a combination of $\mathrm{Indx}(t)$ and shipping quantities is chosen from the list $L(t, \tau_{\kappa-1}, \kappa, I_\kappa(t-1))$ of all possible allocations (into specific items), as explained above.

Given the choice of an element from the list $L(\cdot)$, the resulting cost has several components. The first term on the right-hand side of Equation (1c) is associated with the holding cost of items that are shipped in period t. Note that according to the accounting mechanism defined, the holding cost is charged to items from the period in which they are shipped (unless consumed immediately), until the period in which they are consumed. In period t this includes items $\kappa, \ldots, \mathrm{Indx}(t)$ because no items of $1, \ldots, \kappa-1$ are shipped in period t and no items of $\mathrm{Indx}(t)+1, \ldots, M$ are kept in inventory at the end of period t.

Next is the holding cost associated with the shipment in periods $t+1, \ldots, \tau_{\kappa-1}-1$, which again consists of several expressions. At the beginning of period $t+1$, given all τ values, the least expensive item shipped in period $t+1$ is $\kappa(t+1)$, and the inventory of items $\kappa(t+1)+1, \ldots, M$ is zero. Moreover, when item $\kappa(t+1)-1$ or a less expensive item is shipped (i.e., in period $\tau_{\kappa(t+1)-1}$), the starting inventory of all items $\kappa(t+1), \ldots, M$ is zero. Therefore the holding cost incurred due to the shipment of items $\kappa(t+1), \ldots, M$ in periods $t+1, \ldots, \tau_{\kappa(t+1)-1}-1$ is represented by the term $G(t+1, \tau_{\kappa(t+1)-1}, \kappa(t+1), I_{\kappa(t+1)}(t))$. Similarly, the holding cost incurred in periods $\tau_{\kappa(t+1)-1}, \ldots, \tau_{\kappa(t+1)-2}-1$ due to the shipment of items $\kappa(t+1)-1, \ldots, M$ in these periods is represented by the term $G(\tau_{\kappa(t+1)-1}, \tau_{\kappa(t+1)-2}, \kappa(t+1)-1, I_{\kappa(t+1)-1}(t) - D_{\kappa(t+1)-1}(t+1, \tau_{\kappa(t+1)-1}))$, and so on (recall that the τ values are nonincreasing). That is, the entire interval considered is divided into subintervals according to the τ values.

As an example, consider a period t with $\kappa = 3$ and associated values of $I_3(t-1)$ and τ_2. Furthermore, assume an element of the list $L(t, \tau_2, 3, I_3(t-1))$, in which $\mathrm{Indx}(t) = \kappa(t+1) = 5$, and some $\tau_3 \ge \tau_4$, as well as $I_3(t)$, $I_4(t)$ and $I_5(t)$ values. Then, according to Equation (1c), the following cost is incurred:

$$
\begin{aligned}
&H(t, I_3(t-1), 5, I_3(t), I_4(t), I_5(t)) + G(t+1, \tau_4, 5, I_5(t)) \\
&\quad + G(\tau_4, \tau_3, 4, I_4(t) - D_4(t+1, \tau_4)) \\
&\quad + G(\tau_3, \tau_2, 3, I_3(t) - D_3(t+1, \tau_3)),
\end{aligned}
$$

where the terms in the above expression represent the holding costs incurred from shipping in period t, periods $t+1, \ldots, \tau_4-1$, periods $\tau_4, \ldots, \tau_3-1$, and periods $\tau_3, \ldots, \tau_2-1$, respectively. (See Figure 1.)

The figure illustrates a decomposition of a subproblem, which consists of periods $t, t+1, \ldots, \tau_2-1$ and items 3, 4, …, M. According to the figure, inventory exists for Items 1 and 2 till the end of the subproblem, so they are not shipped during these periods. Item 3 is the least expensive item shipped in period t, and the dotted line corresponding to it represents its stock at the end of period t. Similarly, the dotted lines corresponding to Items 4 and 5 represent the stock of those items at the end of period t (shown as the

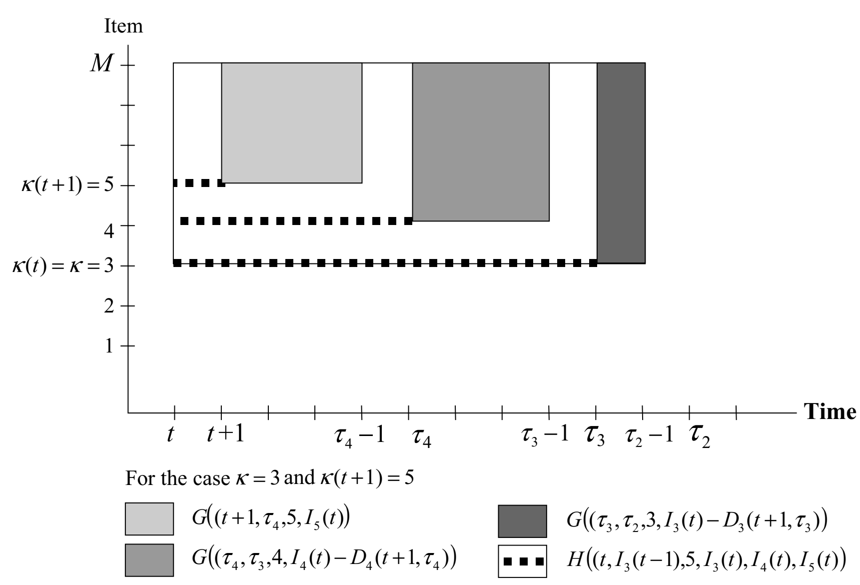

Figure 1 **Decomposition of** $G(t, \tau_2, 3, I_3(t-1))$

number of future time periods covered by this stock). No units of items $6, \ldots, M$ are held in stock at the end of period t: the quantities shipped of these items in period t were just enough for the consumption in period t. As a result, the holding cost attributed to period t is the cost of holding the stock of only Items 3, 4, and 5 (the dotted lines), represented by the H function.

We also observe from the figure that while Item 5 is shipped again in the next period (period $t+1$), the stock of Item 3 covers demand for this item up to the end of period $\tau_3 - 1$, and therefore the next time Item 3 is shipped is in period τ_3. Similarly, Item 4 is not shipped until period τ_4. Thus, the strategy in period t results in the decomposition of the subproblem into three subsubproblems: (i) periods $t+1, \ldots, \tau_4 - 1$ with items $5, 6, \ldots, M$; (ii) periods $\tau_4, \ldots, \tau_3 - 1$ with items $4, 5, \ldots, M$; (iii) periods $\tau_3 - 1, \ldots, \tau_2 - 1$ with items $3, \ldots, M$. These three subsubproblems correspond to the shaded rectangles in the figure, and the total holding cost attributed to items shipped in their associated periods is represented by the appropriate G function. Note that in each subproblem the only item that can be held in stock at the beginning of the first period is of the least expensive item, and moreover, at the end of the last period of the subproblem the stock of all items corresponding to the subproblem vanish.

The $G(\cdot)$ functions are evaluated recursively, until a resulting $G(\cdot)$ function is trivial to compute.

Order of Computation

The value of $G(t, \cdot)$ is computed for t values going downward from T to $T-1, T-2, \ldots, 2$ (for $t=1$, only the value $G(1, T+1, 1, 0)$ is required). For a given t, $1 \le t \le T$, and a given κ, $1 \le \kappa \le M+1$, the value of $G(t, \tau_{\kappa-1}, \kappa, \cdot)$ is computed for $\tau_{\kappa-1}$ values going from $T+1$ downward to T, $T-1, \ldots, t+1$. For each combination of t and $\tau_{\kappa-1}$, the $G(\cdot)$ value is computed downward in the κ index; that is, from $M+1$ to M, $M-1, \ldots, 1$. For each combination of t, $\tau_{\kappa-1}$ and κ, there are $O(t)$ possible values of $I_\kappa(t-1)$ that need to be considered (in any order), as explained below.

Starting with the last period, we have the first initial condition (assuming $T > 1$):

$$G(T, T+1, \kappa, I_\kappa(T-1)) = \begin{cases} 0 & \text{if } \left((d_{\kappa T} - I_\kappa(T-1))^+ + \sum_{j=\kappa+1}^{M} d_{jT}\right) \bmod C = 0 \\ \infty & \text{otherwise.} \end{cases} \tag{2}$$

This condition determines that shipping may occur in period T only if the remaining net demand is a multiple of C.

The rest of the initial conditions refer to the case $\kappa = M$. This case implies that $\kappa > 1$ and therefore $t > 1$ because for $t = 1$ we only need to compute the value $G(1, T+1, 1, 0)$ in which $\kappa = 1$.

The next condition refers to some earlier period and the most expensive item. For $t < T$,

$$G(t, T+1, M, I_M(t-1)) = \begin{cases} \infty & \text{if } (D_M(t, T+1) - I_M(t-1)) \bmod C \neq 0 \\ H(t, I_M(t-1), M, I_M(t)) + G(t+1, T+1, M, I_M(t)) & \text{if } I_M(t) < d_{M\,t+1} \\ H(t, I_M(t-1), M, I_M(t)) + G(t+1, \tau_M, M+1, 0) + G(\tau_M, T+1, M, I_M(t) - D_M(t+1, \tau_M)) & \text{otherwise,} \end{cases} \tag{3}$$

where $I_M(t) = C - (d_{Mt} - I_M(t-1)) \bmod C$. In the first case of condition (3), shipment cannot occur according to the optimality conditions. In the last two cases, item M is shipped to satisfy period t's demand of this item, and at the end of period t some of the units that were shipped remain in inventory and incur holding costs. In the second case, the inventory of item M at the end of period t does not cover the demand of this item in period $t+1$, therefore item M must be shipped again in period $t+1$. In the last case, item M is not shipped until period $\tau_M > t+1$.

Conditions (2) and (3) may be written in more generality as in conditions (4) and (5):

$$G(t, t+1, \kappa, I_\kappa(t-1)) = \begin{cases} 0 & \text{if } \left((d_{\kappa t} - I_\kappa(t-1))^+ + \sum_{i=\kappa+1}^{M} d_{it}\right) \bmod C = 0 \\ \infty & \text{otherwise.} \end{cases} \tag{4}$$

For $\tau_{M-1} > t+1$,

$$G(t, \tau_{M-1}, M, I_M(t-1)) = \begin{cases} \infty & \text{if } (D_M(t, \tau_{M-1}) - I_M(t-1)) \bmod C \neq 0 \\ H(t, I_M(t-1), M, I_M(t)) + G(\tau_M, \tau_{M-1}, M, I_M(t) - D_M(t+1, \tau_M)) & \text{otherwise,} \end{cases} \tag{5}$$

where again, $I_M(t) = C - (d_{Mt} - I_M(t-1)) \bmod C$.

In the second case of (5) we use $G(t+1, \tau_M, M+1, 0) = 0$, which was discussed earlier and (for completeness) is stated next:

$$G(t, \tau_M, M+1, 0) = 0 \quad \text{for all } t < \tau_M, \tag{6}$$

where the zero in the last element is due to the fact that $I_{M+1}(t-1)=0$.

We now return to the explanation of how to efficiently construct the list $L(\cdot)$.

Constructing the List $L(t, \tau_{\kappa-1}, \kappa, I_\kappa(t-1))$ when $\tau_{\kappa-1} > t+1$

Recall that a member in this list specifies the most expensive item held in inventory at the end of period t (i.e., $\text{Indx}(t)$ where $\text{Indx}(t) \geq \kappa$ is implied by $\tau_{\kappa-1} > t+1$), the least expensive item shipped in period $t+1$ (i.e., $\kappa(t+1)$ where $\kappa(t+1) \geq \text{Indx}(t)$), the inventory levels of all items m, $\kappa \leq m \leq \text{Indx}(t)$ at the end of period t (i.e., $I_\kappa(t), I_{\kappa+1}(t), \ldots, I_{\text{Indx}(t)}(t)$), and the period for each item $i \geq \kappa$, in which it or a less expensive item will be shipped next (denoted as defined above by $\tau_\kappa, \tau_{\kappa+1}, \ldots, \tau_M$). Also recall that the values $\tau_1, \ldots, \tau_{\kappa-1}$ that are already known at the end of period $t-1$, are not affected by the shipment in period t because items $1, \ldots, \kappa-1$ are not shipped in period t.

By definition of the τ values, the value of $\tau_{\kappa-1}$ provides an upper bound for the new determined τ's $(\tau_\kappa, \ldots, \tau_M)$, where $t+1$ provides a lower bound for these new determined τ's. More specifically, $t+1 \leq \tau_M \leq \cdots \leq \tau_\kappa \leq \tau_{\kappa-1}$. For $\tau_{\kappa-1} > t+1$, all sequences of $\tau_\kappa, \tau_{\kappa+1}, \ldots, \tau_M$, that satisfy $t+1 \leq \tau_M \leq \cdots \leq \tau_\kappa \leq \tau_{\kappa-1}$, have to be considered as elements in the list $L(\cdot)$. However, sequences that do not satisfy the optimality conditions are excluded (see the explanation below).

Given a newly determined sequence of $\tau_\kappa, \tau_{\kappa+1}, \ldots, \tau_M$ at the end of period t, the value $\kappa(t+1)$ may be computed as follows: $\kappa(t+1) = \min\{i \geq \kappa: \tau_i = t+1\}$; that is, $\kappa(t+1)$ and the calculation of the $G(t+1, \tau_{\kappa-1}, \cdot, \cdot)$ function is independent of the values of $\tau_1, \tau_2, \ldots, \tau_{\kappa-2}$. Note also that $\kappa(t+1) \geq \text{Indx}(t)$. Thus, a member in the list $L(t, \tau_{\kappa-1}, \kappa, I_\kappa(t-1))$ is a vector $(\kappa, \text{Indx}(t), \kappa(t+1), I_\kappa(t), \ldots, I_{\text{Indx}(t)}(t), \tau_\kappa, \tau_{\kappa+1}, \ldots, \tau_M)$.

We now analyze which elements may enter the list $L(\cdot)$, and show that a feasible specification of $\tau_\kappa, \ldots, \tau_M$ (thus specifying $\kappa(t+1)$), uniquely determines the inventory levels of $I_\kappa(t), \ldots, I_{\text{Indx}(t)}(t)$. Toward that, recall that the shipping quantities in all periods, except possibly for Period 1, are in full vehicles. Therefore, in each period we know a priori the value of $X(t) \bmod C$. In fact, given $I_\kappa(t-1)$, the exact amount of $X(t)$ is known because no more than $C-1$ units are shipped for future periods (see Lemma 1 (P2) and Restriction 4 below). We also note that in each subproblem $G(\tau_i, \tau_{i-1}, i, I_i(\tau_i-1))$, $\kappa \leq i \leq \kappa(t+1)-1$ in Equation (1c), we consider periods $\tau_i, \tau_i+1, \ldots, \tau_{i-1}-1$ and items $i, \ldots, M$ only. That is, we start the subproblem in period τ_i with $I_i(\tau_i-1)$ units of item i and no inventory of items $i+1, \ldots, M$. The subproblem ensures that we start period τ_{i-1} (i.e., the next subproblem) with no inventory of items $i, i+1, \ldots, M$, i.e., all $I_i(\tau_i-1)$ units of item i that are held in inventory at the beginning of period τ_i, as well as all items that are shipped in periods $\tau_i, \ldots, \tau_{i-1}-1$, are consumed before period τ_{i-1}. Therefore,

$$\left(\sum_{j=i}^{M} D_j(\tau_i, \tau_{i-1}) - I_i(\tau_i-1)\right) \bmod C = 0 \quad \text{for } \tau_i > 1. \tag{7}$$

Applying condition (7) recursively, from item $i=M$ backward to item $i=\kappa$, results in a condition on $I_i(\tau_i-1) \bmod C$ for all $\kappa \leq i \leq M$ (see Restriction 1 below).

Then, note that for item $i > \text{Indx}(t-1)$, the inventory at the beginning of period t is zero, and therefore at the end of period t we have

$$\begin{cases} I_i(t) < C & \text{if } i > \text{Indx}(t-1) \\ I_i(t) - (I_i(t-1) - d_{it})^+ < C & \text{if } \kappa = \text{Indx}(t-1) \\ & \quad \text{and } i = \kappa. \end{cases} \tag{8}$$

(See Lemma 1 (P2).) Condition (8) is associated with Restriction 2 below.

Thus, given at the end of period t, a sequence τ_i, $\kappa \leq i \leq \text{Indx}(t)$, and the value of $\tau_{\kappa-1}$, we obtain four restrictions on the value of $I_i(t)$:

Restriction 1. Its $\bmod C$ value is restricted to ensure that the net demand of all items $i, \ldots, M$ in periods $\tau_i, \ldots, \tau_{i-1}-1$, which is the same as the total shipment quantity in these periods, obeys condition (7).

Restriction 2. Given $I_\kappa(t-1)$, the value of $I_i(t)$ for $\kappa \leq i \leq \text{Indx}(t)$ is restricted to at most C consecutive values (see Lemma 1 (P2)). (Indeed, we use here a relaxed version of this property. The property states that we never dispatch a vehicle whose entire content is for future periods. Here, we say that we never dispatch a vehicle of a specific item whose entire content is for future periods.)

Restriction 3. The value $I_i(t)$ should also satisfy $I_i(t) \geq D_i(t+1, \tau_i)$ and $I_i(t) \leq D_i(t+1, \tau_{i-1})$ for $\kappa \leq i \leq \text{Indx}(t)$.

Restriction 4. Invoking again Lemma 1 (P2), we obtain a restriction on

$$I_\kappa(t) - (I_\kappa(t-1) - d_{\kappa t})^+ + \sum_{i=\kappa+1}^{\text{Indx}(t)} I_i(t),$$

which represents the total shipment quantity in period t for use in future periods. This amount is restricted to be less than C. Thus, in addition to the first three restrictions on the value of specific $I_i(t)$ values, this additional restriction refers to a sum of inventory levels.

The fifth restriction is associated with the most expensive item only:

Restriction 5. $I_M(t) < C$ for all $1 \le t \le T$, that is, the amount carried in inventory from the most expensive item is always less than C (see Lemma 1 (P4)).

Thus, Restrictions 1 and 2 uniquely specify the possible value of $I_i(t)$ for $\kappa \le i \le \mathrm{Indx}(t)$, given a sequence τ_i, $\kappa \le i \le \mathrm{Indx}(t)$. The modulo C value of $I_i(t)$ is found by solving condition (7), and its precise value is found by also considering condition (8). Restrictions 3 and 4 impose further conditions on the above inventory values, and may indicate in some cases that the sequence τ_i, $\kappa \le i \le \mathrm{Indx}(t)$, cannot satisfy the optimality conditions and thus should not be considered in the list $L(\cdot)$. Restriction 5 does the same with respect to the most expensive item.

One can observe that the number of possible combinations of inventory and τ values is small. As discussed above, every sequence of τ values determines at most one possible choice of the set of inventory values. Hence, when counting the number of elements in the list $L(\cdot)$, the number of possible sequences of inventory values may be ignored. We consider now the rest of the parameters.

In a given period t, for every item, there are $T - t$ possible values of τ and therefore the number of possible sequences of the τ values is $O((T-t)^{M-\kappa+1}) = O(T^M)$. The rest of the parameters in an element of the list $L(\cdot)$ are uniquely determined by the sequence of the τ values, thus the number of elements in a given list $L(\cdot)$ is also $O(T^M)$. To obtain the complexity of the entire dynamic programming algorithm, the number of times that Equation (1) has to be solved remains to be determined. Here we note that for a given subproblem, Equation (1) has to be solved for every combination of t, $\tau_{\kappa-1}$, κ and $I_\kappa(t-1)$ values. The parameters t, $\tau_{\kappa-1}$ and κ create a total of $O(T^2M)$ combinations. In addition, the parameter $I_\kappa(t-1)$ may receive $O(t) = O(T)$ values because its $\bmod C$ value is determined by (7), and its value is bounded by $t \cdot C$, because in every period no more than C units may be stored for future periods (see Lemma 1 (P2)). Hence, there are a total of $O(T^3M)$ combinations of t, $\tau_{\kappa-1}$, κ and $I_\kappa(t-1)$ values, and the complexity of solving a given subproblem is $O(T^{M+3}M)$. Because there are $O(T^2)$ subproblems (arcs in the shortest-path network—see §2), the complexity of solving the entire problem is $O(T^{M+5}M)$.

In other words, for a given number of items, M, the complexity of solving the problem is polynomial! Only when M is part of the input is the complexity of our dynamic programming algorithm exponential. For the latter case, we conjecture that the problem is indeed NP-hard, although we were not able to prove it formally.

A detailed example that illustrates the dynamic programming solution procedure is provided in Appendix C.

4. Conclusions

Most of the literature on transportation problems focuses on the single-item case. It is usually complex to generalize results obtained for the single-item case to settings with multiple items of different characteristics. For example, in capacitated shipments, if the items differ in their volume or weight specifications, or both, it is likely that the solution method for the problem would have to include or be combined with a bin-packing procedure. In this paper we consider items that differ in one important characteristic: the holding cost rate. The items are identical in all other aspects, such as size, volume, transportation requirements (e.g., temperature), and so on.

Our problem deals with shipments between two facilities, therefore the transportation cost of dispatching a vehicle consists of a fixed cost plus variable item-dependent transportation cost, where the latter term has no impact on the solution. While the SIMV version of this problem is known to be polynomially solvable even for the nonstationary cost coefficients case, the complexity of the MIMV problem is unknown, even for the simplest case of stationary cost coefficients. We investigate the structure of an optimal policy and partially unveil the complexity issue of the MIMV problem for stationary cost coefficients by proving that the problem is polynomially solvable for a fixed number of items. The dynamic programming formulation that we propose may be used to solve optimality problems with a small number of items, due to the high complexity when the number of items increases. An alternative solution method is needed for problems with a large number of items. We are currently working on developing such a procedure.

Acknowledgments
The authors would like to thank Gilbert Laporte, the former editor of the journal, and the anonymous referees for the efficient handling of the paper and the helpful comments.

Appendix A. The Scheduling Algorithm

We give here a formal description of the Scheduling algorithm, which, given a vector $(X(1), \ldots, X(T))$ of aggregated quantities shipped in each period, computes the best detailed schedule for each period. In addition, the algorithm performs a test, checking whether any of properties (P1)–(P4) in Lemma 1 is violated. If any of them is violated, then we say that the given vector is not a candidate to be an optimal schedule for periods $1, \ldots, T$. In such a case the algorithm returns a message that the schedule is not optimal. Finally, the algorithm performs a feasibility test, and returns a message of infeasibility if there exists a period up to which the cumulative aggregated demand exceeds the cumulative aggregated shipment quantity.

```
The Scheduling Algorithm
Input: X(t) 1 ≤ t ≤ T
Output: an infeasibility message, or the best-detailed
   schedule and its cost or a message that the schedule
   cannot be optimal.
Feasibility Test: (A feasible schedule exists if and only if
   Σ_{τ=1}^{t} X(τ) ≥ D(1,t+1) for 1 ≤ t ≤ T−1 and Σ_{t=1}^{T} X(t) =
   D(1,T+1).)
t = 1
while t ≤ T−1 do begin
      if Σ_{τ=1}^{t} X(τ) < D(1,t+1) then
         goto (s)        (no feasible solution exist)
      t = t+1
endwhile
If Σ_{τ=1}^{T} X(τ) ≠ D(1,T+1) then
   goto (s)         (no feasible solution exist)

Construction of the Best Detailed Schedule:
Begin                       (initialization step)
I(0) = 0
for t = 1,...,T
      I(t) = 0
      for i = 1,...,M
            d'_{it} = d_{it}
            I_i(t) = 0
      endfor;
endfor;
V = K Σ_{t=1}^{T} ⌈X(t)/C⌉      (current cost consists of
                                 dispatching costs only)
t ← T
(b)   W ← X(t)
      F = 0       (F counts the number of units shipped
                   in each period for future periods)
      i ← M

(a)   X_i(t) ← min{d'_{it}, W}
      If X_i(t) < W then begin
            W ← W − X_i(t)
            i ← i−1
            goto (a)
      endif
      otherwise do begin

            if t > 1 do begin
                  d'_{i(t−1)} = d'_{i(t−1)} + d'_{it} − W
                     (shift the remaining demand of i
                      from period t to period t−1)
                  I_i(t−1) = d'_{it} − W
                  F = F + I_i(t) − max{0, I_i(t−1) − d_{it}}
                  I(t−1) = I(t−1) + I_i(t−1)
                  V = V + h_i I_i(t−1)   (add holding cost
                                          of i from t−1 to
                                          t to V)
                  if i > 1 do begin
                        for k = 1,...,i−1 do begin
                              I_k(t−1) = d'_{kt}
                              d'_{k(t−1)} = d'_{k(t−1)} + d'_{kt}
                              F = F + I_k(t)−
                                 max{0, I_k(t−1) − d_{kt}}
                              I(t−1) = I(t−1) + I_k(t−1)
                              V = V + h_k I_k(t−1)
                                 (add to V the holding
                                  cost of item k from
                                  t−1 to t)
                        endfor;
                  endif;
            endif;

      endotherwise
      if t = 1 then F = I(1);
      if [I(t−1) > 0 and X(t) mod C > 0] or F ≥ C or
            I_M(t) ≥ C or (F ≥ X(t) mod C and
            X(t) mod C > 0) or [I(t−1) = 0 and
            Σ_{i=1}^{M} d_{it} mod C = 0 and I(t) > 0]
            then do begin
            "it is not an optimal solution"
            stop.
      endif;

      t ← t−1
      if t = 0 stop (the best allocation is found)
      goto (b)
(s)   "no feasible solution exists"
      endAlgorithm
```

Appendix B. The Shifting Algorithm

```
t ← T
while t > 1 do begin
      if D(t,t+1) ≤ C then
            for i = 1,...,M D'_i(t,t+1) ← D_i(t,t+1)
      otherwise do begin
            C' ← C
            i ← M+1
            while C' > 0 do begin
                  i ← i−1
                  D'_i(t,t+1) = min{C', D_i(t,t+1)}
                  C' = C' − D'_i(t,t+1)
            endwhile
            D_i(t−1,t) ← D_i(t−1,t) + {D_i(t,t+1)−
               D'_i(t,t+1)}
            for k = 1,...,i−1 do begin
                  D'_k(t,t+1) = 0
                  D_k(t−1,t) ← D_k(t−1,t)+
                     D_k(t,t+1)
            endfor;
      D(t,t+1) ← Σ_{i=1}^{M} D_i(t,t+1)
      endotherwise;
      t ← t−1
endwhile;
```

Appendix C

We illustrate the dynamic programming procedure by an example.

Example: A problem of three periods and three items (this problem could be an arc in the network of a larger problem). The demand is given in the next table:

	Period		
Item	1	2	3
3	4	3	8
2	5	7	5
1	2	2	18

Other data are $C = 10$, $h_1 = 1$, $h_2 = 2$, $h_3 = 3$. Total demand in this problem is 54 units.

Solving for $t = T = 3$

Set $\tau_{\kappa-1} = T + 1 = 4$. This is the only value of $\tau_{\kappa-1}$ that has to be considered.

We apply condition (4), from the most to the least expensive item.

For $\kappa = 3$, we get:

$$G(3, 4, 3, I_3(2)) = \begin{cases} 0 & \text{if } ((d_{33} - I_3(2))^+) \bmod 10 = 0 \\ \infty & \text{otherwise.} \end{cases}$$

Note that $I_3(2) \le d_{33} = 8$ (no positive ending inventory),
$\Longrightarrow$ $G(3, 4, 3, 8) = 0$ and $G(3, 4, 3, I_3(2)) = \infty$ for $I_3(2) \ne 8$.

For $\kappa = 2$, we get:

$$G(3, 4, 2, I_2(2)) = \begin{cases} 0 & \text{if } ((d_{23} - I_2(2))^+ + d_{33}) \bmod 10 = 0 \\ \infty & \text{otherwise.} \end{cases}$$

Thus, because $d_{23} = 5$ and $d_{33} = 8$, the expression is zero if $((5 - I_2(2))^+ + 8) \bmod 10 = 0$, which is equivalent to $(5 - I_2(2))^+ \bmod 10 = 2$, which is satisfied only for $I_2(2) = 3$.

The above calculation is in fact equivalent to the following argument: Because the demand that needs to be covered is 13 units, $I_2(2) \bmod 10 = 3$. Then, because the demand of Item 2 that needs to be covered is five units, $I_2(2) < 5$, which leaves only one possibility: $I_2(2) = 3$.
$\Longrightarrow$ $G(3, 4, 2, 3) = 0$ and $G(3, 4, 2, I_2(2)) = \infty$ for $I_2(2) \ne 3$.

For $\kappa = 1$, we get:

$$G(3, 4, 1, I_1(2)) = \begin{cases} 0 & \text{if } ((d_{13} - I_1(2))^+ \\ & \quad + d_{23} + d_{33}) \bmod 10 = 0 \\ \infty & \text{otherwise.} \end{cases}$$

That is, the total demand in this subproblem is 31 units, therefore $I_1(2) \bmod 10 = 1$. Then, because the demand of Item 1 in this subproblem is 18 units, $I_1(2) < 18$, there are two possibilities: $I_1(2) = 1$ or $I_1(2) = 11$.

For $I_1(2) = 1$, the net demand in Period 3 is 30. Therefore, we ship 30 units, clearly with no units left in inventory. $\Rightarrow$ $G(3, 4, 1, 1) = 0$.

For $I_1(2) = 11$, the net demand in Period 3 is 20. Therefore, we ship 20 units, with no units left in inventory. $\Rightarrow$ $G(3, 4, 1, 11) = 0$.

Also, $G(3, 4, 1, I_1(2)) = \infty$ for $I_1(2) \ne 1$ or 11.

Solving for $t = 2$

Set first $\tau_{\kappa-1} = 4$.

For $\kappa = M = 3$, apply (5) and obtain:

$$G(2, 4, 3, I_3(1)) = \begin{cases} \infty \quad \text{if } \left(\sum_{\tau=2}^{3} d_{3\tau} - I_3(1)\right) \bmod 10 \ne 0 \\ H(2, I_3(1), 3, I_3(2)) \\ \quad + G(\tau_3, 4, 3, I_3(2) \\ \qquad - D_3(3, \tau_3)) \qquad \text{otherwise.} \end{cases}$$

The demand of Item 3 in this subproblem is 11 units, therefore $I_3(1) \bmod 10 = 1$, and $I_3(1) < 11$, which leaves only one possibility: $I_3(1) = 1$. The shipping quantity will be of 10 units, implying $\tau_3 = 4$, $I_3(2) = 8$ and $I_3(2) - D_3(3, 4) = 0$.
$\Longrightarrow$ $G(2, 4, 3, 1) = H(2, 1, 3, 8) = 24$, and $G(2, 4, 3, I_3(1)) = \infty$ for $I_3(1) \ne 1$.

For $\kappa = 2$ we use the general functional Equation (1) and obtain

$$G(2, 4, 2, I_2(1)) = \infty \quad \text{if } (23 - I_2(1)) \bmod 10 \ne 0,$$
$$\Longrightarrow \quad G(2, 4, 2, I_2(1)) = \infty \quad \text{for } I_2(1) \bmod 10 \ne 3.$$

For $I_2(1) \bmod 10 = 3$, we apply the minimization of (1c).

Out of the 23 units of this subproblem, only 12 are of Item 2. Therefore, $I_2(1) < 12$, which combined with $I_2(1) \bmod 10 = 3$ leaves only the possibility $I_2(1) = 3$. The net demand in Period 2 is 7 units, therefore we ship 10 units, and 3 units of Item 2 remain in inventory. (Because the amount of inventory is smaller than the next period's demand of the least expensive item shipped, then by Lemma 2 all the inventory will be of the least expensive item shipped.)

The element in the list $L(2, 4, 2, 3)$ that is associated with this shipping schedule is $\kappa = 2$, $\tau_2 = 3$, $I_2(2) = 3$, $\text{Indx}(2) = 2$, and $\kappa(3) = 2$. That is, $L(2, 4, 2, I_2(1)) = \{(2, 2, 2, 3, 3, 3)\}$, and the associated cost is

$$\begin{aligned} G(2, 4, 2, 3) &= H(2, I_2(1) = 3, \text{Indx}(2) = 2, I_2(2) = 3) \\ &\quad + G(3, \tau_1 = 4,\ 2,\ I_2(2) = 3) \\ &= H(2, 3, 2, 3) + 0 = 6. \end{aligned}$$

Note that $\tau_1 = 4$ by definition of this case, since $\kappa = 2$ and $\tau_{\kappa-1} = 4$.

For $\kappa = 1$ we obtain:

Demand in this subproblem is 43 units $\Rightarrow I_1(1) \bmod 10 = 3$.

Demand of Item 1 that needs to be covered: 20 units $\Rightarrow I_1(1) < 20 \Rightarrow I_1(1) = 3$ or 13.

$\Longrightarrow$ $G(2, 4, 1, I_1(1)) = \infty$ for $I_1(1) \ne 3$ or 13, and we compute $G(2, 4, 1, 3)$ and $G(2, 4, 1, 13)$.

For $I_1(1) = 3$, the net demand in the period is 10. Therefore, we ship 10 units, 7 units of Item 2 and 3 units of Item 3, and no units of Item 1. Therefore, this is a contradiction to $\kappa = 1$, so this case is not possible.

The case $I_1(1) = 13$ is impossible because by Lemma 1 (P2) the maximum inventory at the end of period t is $t \cdot (C - 1)$, i.e., $I_1(1) \le 9$.

$\Longrightarrow$ $G(2, 4, 1, I_1(1)) = \infty$ for all values of $I_1(1)$.

Now set $\tau_{\kappa-1} = 3$ and apply (4).

For $\kappa = 3$:

$$G(2,3,3,I_3(1)) = \begin{cases} 0 & \text{if } (d_{32} - I_3(1))^+ \bmod 10 = 0 \\ \infty & \text{otherwise.} \end{cases}$$

$\Longrightarrow$ $G(2,3,3,I_3(1)) = \infty$ for $I_3(1) \neq 3$. For $I_3(1) = 3$ note that at the end of Period 1 $\tau_3 = 3$, contradicting $\kappa = 3$ for Period 2. Thus, $I_3(1) > 0$ is impossible.

For $\kappa = 2$:

$$G(2,3,2,I_2(1)) = \begin{cases} 0 & \text{if } ((d_{22} - I_2(1))^+ + d_{32}) \bmod 10 = 0 \\ \infty & \text{otherwise.} \end{cases}$$

$\Longrightarrow$ We need $((7 - I_2(1))^+ + 3) \bmod 10 = 0$, which is equivalent to $(7 - I_2(1))^+ \bmod 10 = 7$.

$\Longrightarrow$ $G(2,3,2,0) = 0$ and $G(3,4,2,I_2(1)) = \infty$ for $I_2(1) > 0$. This is an example for $\kappa(t) = \kappa(2) > \text{Indx}(1) = \text{Indx}(t-1)$.

For $\kappa = 1$:

$$G(2,3,1,I_1(1)) = \begin{cases} 0 & \text{if } ((d_{12} - I_1(1))^+ + d_{22} + d_{32}) \bmod 10 = 0 \\ \infty & \text{otherwise.} \end{cases}$$

$\Longrightarrow$ We need: $((2 - I_1(1))^+ + 10) \bmod 10 = 0$, which is equivalent to $(2 - I_1(1))^+ \bmod 10 = 0$. This implies that $I_1(1) = 2$, but in this case we ship 10 units of Items 2 and 3 and no units of Item 1, contradicting the fact that $\kappa = 1$. Therefore, $G(2,3,1,I_1(1)) = \infty$ for any value of $I_1(1)$.

Solving for $t = 1$

(We need only the value G(1,4,1,0).) Demand that needs to be covered: 54 units $\Rightarrow$ the shipping quantity in Period 1 has *modulo* $10 = 4$, and because $I_1(0) = 0$ and the demand in Period 1 is 11, the exact amount shipped is 14.

As a result, three units are kept in inventory. To account for the possible allocations (into items) of these units, we may consider all possible τ sequences, where $\kappa = 1$ hence $\tau_{\kappa-1} = \tau_0 = T + 1 = 4$, and $t + 1 = 2 \leq \tau_3 \leq \tau_2 \leq \tau_1 \leq \tau_0 = 4$.

However, it is clear here that out of these three units, two units are necessarily of Item 1 (see Lemma 2), and the additional unit is either of Item 1 or 2.

Therefore, there are two cases to consider:

- $\tau_1 = 3$, $\tau_2 = 2$, and $\tau_3 = 2$; therefore $\kappa(2) = 2$. In this case, Restriction (3) implies $I_3(1) = 0$, and Equation (7) implies that $(d_{22} + d_{32} - I_2(1)) \bmod C = 0$. Therefore, $(10 - I_2(1)) \bmod C = 0$, thus $I_2(1) = 0$, implying $I_1(1) = 3$.
- $\tau_1 = 2$, $\tau_2 = 2$, and $\tau_3 = 2$. Therefore, $\kappa(2) = 1$. Restriction 3 implies $I_3(1) = I_2(1) = 0$, thus $I_3(1) = 0$. However, the net demand in Period 2 is 10, and therefore no units of Item 1 are shipped in Period 2, a contradiction to $\kappa(2) = 1$. Therefore, this option is not possible.

Therefore, the list $L(1,4,1,0)$ contains one element:

$$L(1,4,1,0) = \{(1,1,2,3,3,2,2)\}$$

and the cost is:

$$\begin{aligned} G(1,4,1,0) &= H(1,0,1,3) + G(2,3,2,0) + G(3,4,1,1) \\ &= 4 + 0 + 0 = 4. \end{aligned}$$

To summarize, the optimal shipping schedule is the following:

	Period		
Item	1	2	3
3	4	3	8
2	5	7	5
1	5 (3 in inventory)	0 (1 in inventory)	17
Total shipment	14	10	30
Total holding costs	3	1	0

In addition to the holding cost of 4, the dispatching cost in the amount of $(2 + 1 + 3)K = 6K$ has to be added to the cost.

References

Anily, S., M. Tzur. 2004. Algorithms for the multi-item capacitated dynamic lot-sizing problem. Working paper, Tel Aviv University, Israel.

Baker, K. R., P. Dixon, M. J. Magazine, E. A. Silver. 1978. An algorithm for the dynamic lot-size problem with time-varying production capacity constraints. *Management Sci.* **24** 1710–1720.

Chen, H. D., D. W. Hearn, C. Y. Lee. 1994. A new dynamic programming algorithm for the single item capacitated dynamic lot size model. *J. Global Optim.* **4** 285–300.

Constantino, M. 1998. Lower bounds in lot-sizing models: A polyhedral study. *Math. Oper. Res.* **23**(1) 101–118.

Federgruen, A, J. Meissner, M. Tzur. 2002. Partitioning heuristics for the multi-item capacitated lot-sizing problem. Working paper, Columbia University, New York.

Florian, M., M. Klein. 1971. Deterministic production planning with concave costs and capacity constraints. *Management Sci.* **18** 12–20.

Florian, M., J. K. Lenstra, A. H. G. Rinnooy Kan. 1980. Deterministic production planning: Algorithms and complexity. *Management Sci.* **26** 669–679.

Lee, C. Y. 1989. A solution to the multiple set-up problem with dynamic demand. *IIE Trans.* **21**(3) 266–270.

Lippman, S. 1969. Optimal inventory policy with multiple set-up costs. *Management Sci.* **16** 118–138.

Pochet, Y., L. A. Wolsey. 1993. Lot-sizing with constant batches: Formulation and valid inequalities. *Math. Oper. Res.* **18** 767–785.

Pryor, K., C. C. White, R. Kapuscinski. 2000. Multi-item inventory policies with capacitated delivery vehicles and deterministic demand. Working paper, University of Michigan, Ann Arbor, MI.

Van Hoesel, S. S., A. Wagelmans. 1996. An $O(T^3)$ algorithm for the economic lot-sizing problem with constant capacities. *Management Sci.* **42** 142–150.

Yano, C. A., A. M. Newman. 2001. Scheduling trains and containers with due dates and dynamic arrivals. *Transportation Sci.* **35**(2) 181–191.