



TI 2002-066/4
Tinbergen Institute Discussion Paper

Polynomial Time Algorithms for Some Multi-Level Lot-Sizing Problems with Production Capacities

Stan van Hoesel¹

H. Edwin Romeijn²

Dolores Romero Morales¹

Albert P.M. Wagelmans³

¹ Faculty of Economics and Business Administration, Maastricht University,

² Department of Industrial and Systems Engineering, University of Florida,

³ Econometric Institute, Faculty of Economics, Erasmus University Rotterdam, and Tinbergen Institute

Tinbergen Institute

The Tinbergen Institute is the institute for economic research of the Erasmus Universiteit Rotterdam, Universiteit van Amsterdam and Vrije Universiteit Amsterdam.

Tinbergen Institute Amsterdam

Keizersgracht 482
1017 EG Amsterdam
The Netherlands
Tel.: +31.(0)20.5513500
Fax: +31.(0)20.5513555

Tinbergen Institute Rotterdam

Burg. Oudlaan 50
3062 PA Rotterdam
The Netherlands
Tel.: +31.(0)10.4088900
Fax: +31.(0)10.4089031

Most TI discussion papers can be downloaded at
<http://www.tinbergen.nl>

Polynomial time algorithms for some multi-level lot-sizing problems with production capacities

Stan van Hoesel* H. Edwin Romeijn^{†‡} Dolores Romero Morales^{§‡}
Albert P.M. Wagelmans[¶]

June 17, 2002

Abstract

We consider a model for a serial supply chain in which production, inventory, and transportation decisions are integrated, in the presence of production capacities and for different transportation cost functions. The model we study is a generalization of the traditional single-item economic lot-sizing model, adding stationary production capacities at the manufacturer, as well as multiple intermediate storage levels (including the retailer level), and transportation between these levels. Allowing for general concave production costs and linear holding costs, we provide polynomial time algorithms for the cases where the transportation costs are either linear, or are concave with a fixed-charge structure. In the latter case, we make the additional common and reasonable assumption that the variable transportation and inventory costs are such that holding inventories at higher levels in the supply chain is more attractive from a variable cost perspective. The running times of the algorithms are remarkably insensitive to the number of levels in the supply chain.

Keywords: Lot-sizing; Integration of production planning and transportation; Dynamic programming; Polynomial time algorithms.

*Faculty of Economics and Business Administration, Maastricht University, P.O. Box 616, 6200 MD Maastricht, The Netherlands; e-mail: s.vanhoesel@ke.unimaas.nl.

[†]Department of Industrial and Systems Engineering, University of Florida, 303 Weil Hall, P.O. Box 116595, Gainesville, Florida 32611-6595; email: romeijn@ise.ufl.edu.

[‡]The work of this author was supported by the National Science Foundation under Grant No. DMI-0085682, Maastricht Research School of Economics of Technology and Organizations (METEOR), and the Netherlands Organization for Scientific Research (NWO).

[§]Faculty of Economics and Business Administration, Maastricht University, P.O. Box 616, 6200 MD Maastricht, The Netherlands; e-mail: d.romero@ke.unimaas.nl.

[¶]Econometric Institute, Erasmus University Rotterdam, P.O. Box 1738, 3000 DR Rotterdam, The Netherlands; e-mail: wagelmans@few.eur.nl.

1 Introduction

In this paper we consider a problem in which production, inventory, and transportation decisions in a basic supply chain are integrated. Traditional models usually consider only one or two of these aspects in isolation from the other(s). Substantial evidence exists (see, for instance, Chandra and Fisher [5], Geoffrion and Powers [11], and Thomas and Griffin [14], as well as the references therein) that shows that integrating these decisions can lead to substantial increases in efficiency and effectiveness. Integrating different decisions in the supply chain are particularly important when resources are limited, and when costs are nonlinear, e.g., exhibit economies of scale. In this paper we will consider a supply chain optimization problem that incorporates both of these aspects.

We will consider a generalization of a fundamental problem, which in fact is one of the most widely studied problems in production and inventory planning, the *economic lot-sizing problem (ELSP)*. The basic variant of this problem considers a production facility that produces and stores a single product to satisfy known demands over a finite planning horizon. The problem is then to determine production quantities for each period such that all demands are satisfied on time at minimal total production and inventory holding costs. The cost functions are non-decreasing in the amount produced or stored, and are usually assumed to be linear, fixed-charge, or general concave functions. The production facility may or may not face a capacity constraint on the amount produced in each period.

When both the production and inventory holding cost functions are concave, the uncapacitated problem is solvable in polynomial time in the length of the time horizon, see Wagner [16] for this basic result. More efficient algorithms for special cases have been developed by Aggarwal and Park [1], Federgruen and Tzur [8], and Wagelmans et al. [15]. When production capacities are present, we obtain the so-called *capacitated lot-sizing problem (CLSP)*. In contrast to the uncapacitated ELSP, this problem is known to be NP-hard, even in many special cases, see Florian et al. [10] and Bitran and Yanasse [4]. An interesting and important special case that does allow for a polynomial time algorithm arises when the production capacities are stationary, see e.g. Florian and Klein [9], Florian et al. [10], and Van Hoesel and Wagelmans [12]. See also references in Baker et al. [3] for other work on the CLSP with stationary production capacities, and Chung and Li [6] for another special case of the capacitated lot-sizing problem that is solvable in polynomial time.

In the context of optimizing the performance of supply chains, it is often relevant to extend the lot-sizing problem to include transportation decisions, as well as the possibility of holding inventory at different levels in the supply chain. For example, consider a manufacturer who produces to satisfy demand at a single retailer over a finite planning horizon of T periods, and suppose that inventory can be held at both the manufacturer and the retailer, although inventory holding costs at the manufacturer and retailer may be different. In particular, this means that items that are produced by the manufacturer may be held in inventory by the manufacturer for a number of periods before being shipped to the retailer. There they may either be used immediately to satisfy demand, or be stored to satisfy future demand. Besides production and inventory holding costs, we then clearly also need to incorporate transportation costs, which adds the problem of the *timing* of transportation to the problem of timing of production. The objective will be to minimize

the system-wide cost while satisfying all demand. Even if the manufacturer and retailer are in fact distinct participants in the supply chain, each of which faces a part of the supply chain costs, this problem will be relevant. In this case, the participants clearly still need to decide how to distribute the minimal total costs, which is a coordination problem that is outside the scope of this paper. But alternatively, we may interpret the holding costs at the retailer level as a penalty or a discount on the purchasing price of an item, which is given by the manufacturer to the retailer if items are delivered early. In this case the costs minimized by our optimization model are all incurred by the manufacturer. As in standard lot-sizing problems, all cost functions are assumed to be non-decreasing in the amount produced, stored, or shipped. In addition, we will also assume that all cost functions are concave or a special case thereof.

The problem sketched above is a two-level lot-sizing problem. In this paper, we will actually study a multi-level version of this problem, in which multiple transportation and storage levels may be present. Zangwill [18] considers an extension of the ELSP to multiple storage levels, and developed a dynamic programming algorithm for uncapacitated multi-level lot-sizing problems. An analysis of this algorithm shows that it runs in $O(LT^4)$ time, where L is the number of levels, or even in $O(T^3)$ for the special case of $L = 2$.

The lot-sizing problem with two or more levels and production capacities is clearly NP-hard, as it is a direct generalization of the CLSP with non-stationary capacities. The main objective of our paper is to identify non-trivial special cases of the multi-level lot-sizing problem that are polynomially solvable in the presence of stationary production capacities. We will study problems with general concave production costs and linear inventory holding costs, and two different transportation cost structures. In particular, we will consider (i) linear transportation costs; and (ii) fixed-charge transportation costs without speculative motives, which means that with respect to variable costs holding inventory is less costly at higher levels than at lower levels in the supply chain. Our polynomial solution methods are based on a dynamic programming framework that uses a decomposition principle that generalizes the classical zero-inventory ordering (ZIO) property of solutions to uncapacitated lot-sizing problems as described in Zangwill [18] for the multi-level case, and, for instance, in Wagner and Whitin [17] for the single-level case.

This paper is organized as follows. In Section 2 we introduce the multi-level lot-sizing problem with general production capacities, and general non-decreasing concave production, transportation, and inventory holding cost functions. We characterize the extreme points of the feasible region of the problem, and prove a decomposition result that will form the basis of our algorithms for several special cases of the problem. In Section 3 we analyze the running time of the method developed by Zangwill [18] for both the two-level and the multi-level version of the problem without production capacities. In Section 4, we study the two-level problem, and provide a general dynamic programming framework based on the decomposition result derived earlier, and develop a polynomial time algorithm for two capacitated models with different transportation cost functions. In Section 5 the algorithms are generalized to the multi-level lot-sizing problem. The paper ends in Section 6 with some concluding remarks and issues for further research.

2 The model

2.1 Introduction

As described above, we will study a multi-level lot-sizing problem with a serial structure. In each period, production may take place at the manufacturer. The items that are produced may be stored at the manufacturer level or transported to the first warehouse level. At each of the warehouse levels, again products are either stored or transported to the warehouse at the next level. From the final warehouse level products are then, after possibly having been stored for some periods, transported to the retailer. Such a structure arises for instance if a retailer actually represents an entire market, and the supply chain from manufacturer to this market is very long. This could make it advantageous to, in several stages, transport larger quantities over long distances to intermediate storage facilities, before being distributed in the actual market. Our model then captures the production and transportation decisions up to a storage facility in the market, but does not take into account the actual distribution to end customers within the market.

We consider a planning horizon of T periods. In each period t , the retailer faces a nonnegative demand given by d_t , while the production capacity of the manufacturer in period t is equal to b_t . We will consider a total of L levels, which includes the manufacturer, the retailer, and $L - 2$ intermediate warehouses. We say that the manufacturer is at the first level of the chain, and the retailer is at the L^{th} level. Each of the intermediary levels corresponds to a warehouse. Let \mathbb{R}_+ denote the set of nonnegative real numbers. The production costs in period t are given by the function $p_t : \mathbb{R}_+ \rightarrow \mathbb{R}_+$. The transportation cost function from level ℓ to level $\ell + 1$ in period t is given by a transportation cost function $c_t^\ell : \mathbb{R}_+ \rightarrow \mathbb{R}_+$, for $\ell = 1, \dots, L - 1$. The inventory holding cost function at level ℓ in period t is given by $h_t^\ell : \mathbb{R}_+ \rightarrow \mathbb{R}_+$, for $\ell = 1, \dots, L$. In the general case, we will assume that all cost functions are concave and non-decreasing.

The *multi-level capacitated lot-sizing problem (MCLSP)* can be formulated as follows:

$$\text{minimize } \sum_{t=1}^T \left(p_t(y_t) + \sum_{\ell=1}^{L-1} c_t^\ell(x_t^\ell) + \sum_{\ell=1}^L h_t^\ell(I_t^\ell) \right)$$

subject to (P)

$$x_t^1 + I_t^1 = y_t + I_{t-1}^1 \quad t = 1, \dots, T \quad (1)$$

$$x_t^\ell + I_t^\ell = x_t^{\ell-1} + I_{t-1}^\ell \quad t = 1, \dots, T; \ell = 2, \dots, L - 1 \quad (2)$$

$$d_t + I_t^L = x_t^{L-1} + I_{t-1}^L \quad t = 1, \dots, T \quad (3)$$

$$y_t \leq b_t \quad t = 1, \dots, T \quad (4)$$

$$I_0^\ell = 0 \quad \ell = 1, \dots, L \quad (5)$$

$$y_t \geq 0 \quad t = 1, \dots, T$$

$$x_t^\ell \geq 0 \quad t = 1, \dots, T; \ell = 1, \dots, L - 1$$

$$I_t^\ell \geq 0 \quad t = 1, \dots, T; \ell = 1, \dots, L$$

where y_t denotes the quantity produced in period t , x_t^ℓ is the quantity shipped from level ℓ to level $\ell + 1$ in period t , and I_t^ℓ denotes the inventory level at level ℓ at the end of

period t . Constraints (1)-(3) model the balance between inflow, storage and outflow at the manufacturer, warehouse, and retailer levels, respectively, in period t . The production quantity in period t is restricted by constraints (4). If $\sum_{\tau=t}^T d_\tau \leq b_t$ for all $t = 1, \dots, T$, the problem is essentially uncapacitated and these constraints are redundant. Constraints (5) impose that the inventory levels at all levels of the chain are equal to zero at the beginning of the planning horizon. As in the standard ELSP, the assumption that the initial inventory level at the retailer level is equal to zero can be made without loss of generality: if this assumption is violated, we may modify the demand data to obtain an equivalent problem with zero initial retailer inventory. The assumption that the initial inventory level at the other levels is zero can also be made without loss of generality when the transportation and inventory costs are linear. In that case, we can simply satisfy earlier demands using these initial inventories, and the corresponding minimal transportation costs can easily be determined. Even in the general case with concave transportation costs having positive initial inventories can be taken into account. However, for ease of exposition, we will assume that the initial inventory at all levels is zero also in this case. We will explicitly discuss how nonzero initial inventory levels can be taken into account in the special case that we will discuss in Section 4.5. Finally, note that due to the monotonicity of the objective function, the inventories at the end of the planning horizon will, without loss of optimality, be equal to zero.

Throughout the paper we will assume that the following condition holds

$$\sum_{\tau=1}^t d_\tau \leq \sum_{\tau=1}^t b_\tau \quad \text{for each } t = 1, \dots, T. \quad (6)$$

Condition (6) states that the aggregate demand in the first t periods can not exceed the total production capacity in these periods. This condition, which is a necessary and sufficient condition for feasibility of the CLSP, is clearly necessary for feasibility of (P) as well. It is easy to see that this condition is also sufficient for feasibility by considering the solution where all demands are satisfied by producing as early as possible.

Proposition 2.1 *Problem (P) is feasible if and only if condition (6) holds.*

The traditional CLSP models only the manufacturer level. It allows for inventories to be held only at that level, and assumes that the transportation costs do not play a role (as is for instance the case if they are both linear and stationary). Our model thus is an extension of the classical CLSP in two directions: it allows for multiple levels at which inventories can be held, and it explicitly incorporates transportation decisions between these levels.

In Section 2.2 we model (P) as a capacitated minimum cost network flow problem in a network with one source (see also Zangwill [18] for a general discussion on minimum costs flows in networks with just one source). In Section 2.3 we characterize the candidate optimal solutions in a way that will allow us to construct polynomial time algorithms for interesting special cases of (P).

2.2 A network representation

Problem (P) can be represented by a network which will be used throughout the paper. The network has a single source S , T transshipment nodes M_t at the production level, T transshipment nodes W_t^ℓ at each of the warehouse levels ($t = 1, \dots, T$; $\ell = 2, \dots, L - 1$), and T demand nodes R_t with demand d_t ($t = 1, \dots, T$) at the retailer level. Feasibility dictates that the source node S has a supply of $\sum_{t=1}^T d_t$ units. Figure 1 illustrates the network representation of (P) for $L = 3$ and $T = 4$.

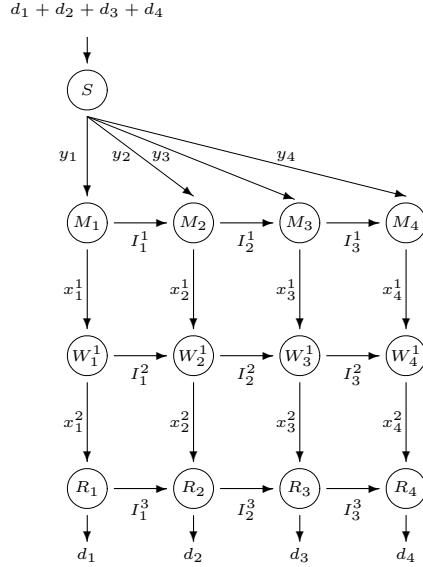


Figure 1: Network representation of (P) for $L = 3$ and $T = 4$.

Note that condition (6) can be interpreted as a cut condition in this network. Consider the set of nodes $\{M_1, \dots, M_t, W_1^2, \dots, W_t^2, \dots, W_1^{L-1}, \dots, W_t^{L-1}, R_1, \dots, R_t\}$. The total demand in periods $1, \dots, t$, i.e. $\sum_{\tau=1}^t d_\tau$, being the outflow of this set of nodes, should be satisfied by its inflow, which in turn is bounded by $\sum_{\tau=1}^t b_\tau$, the sum of the capacities of the production arcs entering this set of nodes (see Ahuja et al. [2]).

The feasible flows in the network enjoy a property that is common in these types of networks. Note that any feasible flow can be decomposed into flows on individual paths from the source node S to the retailers R_t . The fact that the network is planar implies that these paths can be chosen in such a way that paths may share edges, but no two paths cross. This can easily be seen as follows: if two paths do cross, we can switch the parts of the paths past the nodes where they cross. Repeating this as often as necessary, we obtain a set of paths that do not cross. This property leads directly to the following proposition.

Proposition 2.2 *Without loss of generality, we may assume that in any feasible solution to (P) production in period t is used to satisfy demand in some set of consecutive periods τ_1, \dots, τ_2 (where $t \leq \tau_1 \leq \tau_2$), where the demands in all periods except possibly τ_1 and τ_2 are satisfied fully.*

Figure 2 illustrates the result of Proposition 2.2 for $T = 5$ and a specific feasible production schedule, where the lengths of the bars indicate the production quantities and demands.

y_1		y_2	y_3	
d_1	d_2	d_3		d_4 d_5

Figure 2: Illustration of production and demand when $T = 5$.

2.3 Characterization of the extreme points

Problem (P) has a concave objective function and its feasible region is defined by linear constraints. This implies that there exists an extreme point optimal solution to (P). Consider the flow in the network corresponding to any extreme point feasible solution. As is common in network flow problems, we will call the arcs that carry an amount of flow that is both strictly positive and strictly less than its capacity *free arcs*. It is well-known (see e.g. Ahuja et al. [2]) that the subnetwork containing only the free arcs contains no cycle.

Note that only arcs that have a finite upper bound (which in our case are only the production arcs) may carry flow while they are not free. Removing all production arcs, the network containing all remaining *free arcs* decomposes into a number of connected components (some of which may be isolated nodes). Each connected component containing flow consists of a set of manufacturer nodes $\{M_{\varsigma_{11}+1}, \dots, M_{\varsigma_{12}}\}$, a set of warehouse nodes at each intermediate level $\{W_{\varsigma_{\ell 1}}^{\ell}, W_{\varsigma_{\ell 2}}^{\ell}\}$ ($\ell = 2, \dots, L - 2$), and a set of retailer vertices $\{R_{\varsigma_{L1}+1}, \dots, R_{\varsigma_{L2}}\}$ with $\varsigma_{\ell 1} \leq \varsigma_{\ell+1,1} < \varsigma_{\ell 2} \leq \varsigma_{\ell+1,2}$ for $\ell = 1, \dots, L - 1$. We will call such a connected component a *subplan*. We will represent a subplan by the $2L$ periods that identify it: $((\varsigma_{\ell 1}, \varsigma_{\ell 2})_{\ell=1}^L)$. It will often be convenient to refer to the production and demand periods in a subplan individually, and we will then use $(t_1, t_2, \tau_1, \tau_2) \equiv (\varsigma_{11}, \varsigma_{12}, \varsigma_{L1}, \varsigma_{L2})$. The only flow entering a subplan comes from production arcs associated with the manufacturer nodes in the subplan. The total flow on all these production arcs is used to supply the demand of all retailer nodes in the subplan, i.e., the total demand in periods $\tau_1 + 1, \dots, \tau_2$. Note that if we would allow that $\varsigma_{\ell+1,1} \geq \varsigma_{\ell 2}$ for some ℓ , then the corresponding subplan would contain no paths from the manufacturer to the retailer level, and would thus not be connected or be able to contain any flow. The fact that the extreme flows are acyclic implies that, although there may be multiple production arcs associated with a subplan that carry flow, there is at most one such arc with production below capacity. In other words, there is at most one free production arc entering the subplan. This yields the following generalization of the characterization of extreme points of single-level capacitated lot-sizing problems by Florian and Klein [9].

Proposition 2.3 *Any extreme feasible solution decomposes into subplans with at most one free production arc entering each subplan.*

If the problem is uncapacitated, this proposition implies that only one production arc carrying flow enters each of the subplans. This means that in that case the extreme flows are *arborescent*. In Section 3, we will analyze the running time of the dynamic

programming algorithm proposed for this problem by Zangwill [18] which is based on this property.

For capacitated problems, an extreme flow does not need to be arborescent. However, the property stated in Proposition 2.3 allows us to derive efficient algorithms to calculate the optimal costs of the subproblems corresponding to subplans, and thereby to the entire problem, for interesting special cases.

3 The uncapacitated case

3.1 General case

When the problem is uncapacitated, the properties discussed in the previous section imply that each arc that carries a positive flow is a free arc. Zangwill [18] used this property to show that an extreme feasible solution (y, x, I) induces a so-called *arborescent* flow in the network. Figure 3 shows an example of an arborescent flow.

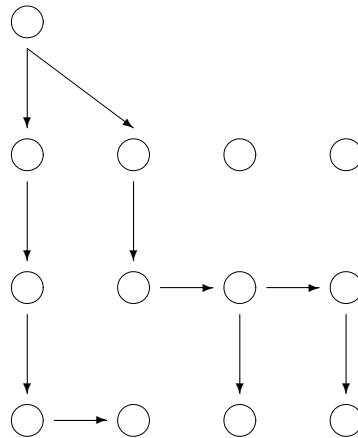


Figure 3: Example of an arborescent flow.

Since an arborescent flow does not contain any undirected cycle, we can conclude that, in an extreme feasible solution, each node in the network has at most one incoming arc carrying flow. An immediate consequence is the standard ZIO property.

Corollary 3.1 *Let (P) have no capacities on production. Then, for an extreme point feasible solution (y, x, I) of (P) we have:*

$$I_t^1 y_{t+1} = 0 \quad t = 1, \dots, T - 1$$

and

$$I_t^\ell x_{t+1}^{\ell-1} = 0 \quad t = 1, \dots, T - 1; \ell = 2, \dots, L.$$

Another important consequence of the structure of the extreme flows concerns the value of the flow on each arc.

Corollary 3.2 *Let (P) have no production capacities. Then an extreme feasible solution (y, x, I) of (P) has the property that each arc that carries positive flow satisfies the entire demand of a consecutive set of periods.*

Proof: Since each node has indegree at most one, this property follows immediately by using backward induction. \square

Zangwill [18] used this property to develop a dynamic programming algorithm to solve the L -level uncapacitated lot-sizing problem. His dynamic programming recursion is a backward recursion in terms of the quantities $C_{t\ell}(s_1, s_2)$, which are defined to be the optimal transportation and inventory costs of shipping $d_{s_1 s_2}$ units from level ℓ at time t to their destinations, i.e., demand nodes s_1, \dots, s_2 , as well as the quantities. The production costs are, in this dynamic programming algorithm, represented at level $\ell = 0$, which essentially means that we have here split the manufacturer into production ($\ell = 0$) and storage ($\ell = 1$) facilities. Clearly, $C_{TL}(T, T) = 0$, and we are interested in $C_{10}(1, T)$. Zangwill's recursions now read

- at the retailer level:

$$C_{tL}(t, s_2) = h_t^L(d_{t+1, s_2}) + C_{t+1, L}(t+1, s_2) \quad t = T-1, \dots, 1; s_2 = T, \dots, t$$

- at the last warehouse level:

$$\begin{aligned} C_{t, L-1}(s_1, s_2) &= h_t^{L-1}(d_{s_1 s_2}) + C_{t+1, L-1}(s_1, s_2) \\ &\quad t = T-1, \dots, 1; s_1 = T, \dots, t+1; s_2 = T, \dots, s_1 \\ C_{t, L-1}(t, s_2) &= \min_{\tau=t, \dots, s_2} \{c_t^{L-1}(d_{t\tau}) + C_{tL}(t, \tau) + h_t^{L-1}(d_{\tau+1, s_2}) + C_{t+1, L-1}(\tau+1, s_2)\} \\ &\quad t = T-1, \dots, 1; s_2 = T, \dots, t \end{aligned}$$

- at the remaining warehouse levels:

$$\begin{aligned} C_{t\ell}(s_1, s_2) &= \min_{\tau=\max(t, s_1-1), \dots, s_2} \{c_t^\ell(d_{t\tau}) + C_{t, \ell+1}(s_1, \tau) + h_t^\ell(d_{\tau+1, s_2}) + C_{t+1, \ell}(\tau+1, s_2)\} \\ &\quad \ell = L-2, \dots, 1; t = T-1, \dots, 1; s_1 = T, \dots, t; s_2 = T, \dots, s_1 \end{aligned}$$

- and at the manufacturer level:

$$C_{t0}(s_1, s_2) = p_t(d_{s_1 s_2}) + C_{t1}(s_1, s_2) \quad t = T-1, \dots, 1; s_1 = T, \dots, t+1; s_2 = T, \dots, s_1.$$

Note that the last warehouse level (which actually coincides with the manufacturer level in case $L = 2$) is treated differently from the remaining warehouse levels. The reason is that the ZIO property implies that if we do not supply demands in the current period, we will not transport to the retailer level. From the above recursions, it is easy to see that the dynamic programming algorithm runs in $O(T^3 + (L-2)T^4)$ time, which boils down to $O(T^3)$ time for $L = 2$, and $O(LT^4)$ for $L > 2$. In Section 5.5 we will use this recursion to determine certain intermediate cost values for a multi-level lot-sizing problem with stationary production capacities.

3.2 A special case

When the number of levels $L = 2$ and the production costs as well as the inventory holding costs at both levels are linear, we can easily reduce the problem to an uncapacitated ELSP. This follows since, given that we decide to transport in a certain period, we can easily determine the best production period, i.e., the period that yields the minimum total unit production and manufacturer-level inventory costs for transport in period t . Redefining the transportation cost function accordingly, which can be done in $O(LT)$ time, allows us to eliminate the production variables as well as the inventory variables at the manufacturer, yielding a standard uncapacitated ELSP. The resulting problem can be solved in $O(T^2)$ time for general concave transportation costs, see Evans [7], and in $O(T \log T)$ time for fixed-charge transportation costs, see Aggarwal and Park [1], Federgruen and Tzur [8], and Wagelmans et al. [15].

4 The two-level capacitated lot-sizing problem

For reasons of clarity, we will first consider the two-level version of the MCLSP, which we will call the 2CLSP. In the next section we will then show how the methodology can be extended to chains with more than two levels.

4.1 A dynamic programming approach

In this section we will outline a general dynamic programming approach to the 2CLSP. To this end, we define $F(t, \tau)$ to be the minimum cost associated with satisfying the retailer demands in periods $\{\tau + 1, \dots, T\}$ using production in periods $t + 1, \dots, T$. We are clearly interested in computing $F(0, 0)$. This can be achieved using the following two-phase approach:

Phase 1: For each subplan $(t_1, t_2, \tau_1, \tau_2)$, compute the minimum costs that are incurred for satisfying the demand of that subplan under the condition that at most one free production arc enters the subplan. Denote these costs by $\phi(t_1, t_2, \tau_1, \tau_2)$.

Phase 2: Compute the values $F(t_1, \tau_1)$ for all $0 \leq t_1 \leq \tau_1 \leq T$ using the following backward recursion:

$$\begin{aligned}
 F(t_1, \tau_1) &= \min_{(t_2, \tau_2): \tau_2 \geq t_2 > \tau_1} \{ \phi(t_1, t_2, \tau_1, \tau_2) + F(t_2, \tau_2) \} && \text{for } 0 \leq t_1 \leq \tau_1 < T \\
 F(t_1, T) &= 0 && \text{for } 0 \leq t_1 \leq T.
 \end{aligned}$$

Note that, in Phase 1, we need to compute $O(T^4)$ values. Phase 2 is in fact a shortest path problem in a network with nodes representing all period-pairs (t, τ) such that $0 \leq t \leq \tau \leq T$, and arcs representing the subplans with corresponding costs. The minimum cost path from node $(0, 0)$ to any of the nodes (t_1, T) in this acyclic network can be found in linear time in the number of arcs, i.e., in $O(T^4)$ time (see Ahuja et al. [2]).

When the value of $\phi(\cdot)$ is given for each subplan, the 2CLSP is polynomially solvable. To achieve a polynomial time algorithm for (P), the challenge is therefore to provide a polynomial time algorithm for computing the costs corresponding to all subplans. Florian

and Klein [9] used this general dynamic programming framework to develop an $O(T^4)$ dynamic programming algorithm to solve the CLSP with stationary capacities and general concave production and inventory holding cost functions.

In the remainder of this section, we will derive a polynomial time algorithm for computing the optimal costs of all subplans, and thereby for (P), for the case of

- stationary production capacities: $b_t = b$ for $t = 1, \dots, T$;
- general concave production costs;
- linear inventory holding costs at both levels

and for two different transportation cost structures:

1. linear transportation costs;
2. fixed-charge transportation costs without speculative motives.

Before studying the subproblems of computing the optimal subplan costs, we will first study the implications of the assumption that the production capacities are stationary as well as the consequences of our cost function choices in more detail in the next sections.

4.2 Implications of stationary production capacities

Recall that in Phase 1 of the dynamic programming approach we need to compute the optimal costs of all subplans, under the additional constraint that all but one of the production arcs entering the subplan carry a flow equal to 0 or b . Now consider a subplan $(t_1, t_2, \tau_1, \tau_2)$, with total corresponding demand equal to $d_{\tau_1+1, \tau_2} = \sum_{\tau=\tau_1+1}^{\tau_2} d_\tau$. Following Florian and Klein [9], note that the constraint on the values of the production arcs entering the subplan implies that the number of production arcs that carry flow equal to the production capacity is exactly equal to

$$K = \left\lfloor \frac{d_{\tau_1+1, \tau_2}}{b} \right\rfloor.$$

The remaining demand is then equal to

$$\varepsilon = d_{\tau_1+1, \tau_2} - Kb$$

and clearly satisfies $0 \leq \varepsilon < b$. If $\varepsilon > 0$, there will be exactly one production arc entering the subplan carrying this flow.

It will be advantageous to compute, in a preprocessing phase of the algorithm, all values $p_t(kb)$ and $p_t(kb + \varepsilon)$ for each pair for each pair (τ_1, τ_2) and all relevant values of t and k . Assuming that a production function evaluation can be done in constant time, this preprocessing step then takes a total of $O(T^4)$ time, which clearly does not dominate the overall running time of the algorithm.

4.3 Consequences of choice of cost functions

So far, we have assumed that all cost functions are concave. However, it is common in lot-sizing problems to model the inventory holding costs as linear functions. Exceptions are Florian and Klein [9], who deal with general concave holding cost functions, and Ortega [13], who deals with fixed-charge inventory holding cost functions. Following most of the literature on lot-sizing problems, we will assume in this paper that the holding cost functions are linear:

$$\begin{aligned} h_t^1(I_t^1) &= h_t^1 I_t^1 \\ h_t^2(I_t^2) &= h_t^2 I_t^2. \end{aligned}$$

Without loss of generality, we can remove linear holding costs from the problem entirely, thereby essentially setting the holding costs equal to zero, by redefining the production and the transportation cost functions. To see this, note that the inventory balance constraints imply that

$$\begin{aligned} I_t^1 &= \sum_{\tau=1}^t y_\tau - \sum_{\tau=1}^t x_\tau \\ I_t^2 &= \sum_{\tau=1}^t x_\tau - \sum_{\tau=1}^t d_\tau. \end{aligned}$$

Substituting these expressions for the inventory variables in the objective function then yields the following redefined production and transportation cost functions:

$$\begin{aligned} \tilde{p}_t(y) &= p_t(y) + \left(\sum_{\tau=t}^T h_\tau^1 \right) y \\ \tilde{c}_t^1(x) &= c_t^1(x) + \left(\sum_{\tau=t}^T h_\tau^2 - \sum_{\tau=t}^T h_\tau^1 \right) x. \end{aligned}$$

These adjusted production and transportation cost functions are clearly still concave.

In Section 4.4, we will assume that the transportation costs are linear, i.e., $c_t^1(x) = g_t x$. Under this choice the redefined transportation cost function will still be linear, which makes it relatively easy to determine, given the production quantities, how demand is satisfied in the most economical way.

In Section 4.5, we will assume that the transportation costs have a fixed-charge structure without speculative motives. More formally,

$$c_t^1(x) = f_t 1_{\{x>0\}} + g_t x$$

where $1_{\{x>0\}}$ is an indicator function taking the value 1 if $x > 0$, and 0 otherwise. The assumption that there are no speculative motives, which is commonly assumed for the production costs in traditional economic lot-sizing models, means that the variable inventory and transportation costs are such that it is attractive to transport as late as possible

when given this choice. Equivalently, holding inventory at the manufacturer level is more attractive than holding inventory at the retailer level:

$$g_t + h_t^2 \geq h_t^1 + g_{t+1} \quad \text{for } t = 1, \dots, T-1.$$

When converting the problem into an equivalent problem with zero inventory costs at both levels as described above, the *no speculative motives* condition becomes a condition on the unit transportation costs as follows:

$$\begin{aligned} g_t + \left(\sum_{\tau=t}^T h_\tau^2 - \sum_{\tau=t}^T h_\tau^1 \right) &= g_t + (h_t^2 - h_t^1) + \left(\sum_{\tau=t+1}^T h_\tau^2 - \sum_{\tau=t+1}^T h_\tau^1 \right) \\ &\geq g_{t+1} + \left(\sum_{\tau=t+1}^T h_\tau^2 - \sum_{\tau=t+1}^T h_\tau^1 \right). \end{aligned}$$

In other words, the (adjusted) variable transportation costs are non-increasing.

Note that if the transportation costs are actually linear, and there are no speculative motives, it is always optimal to store production at the manufacturer and transport only when demand needs to be satisfied. Hence, without loss of optimality, we can assume that $I_t^2 = 0$ for all $t = 1, \dots, T$. Similarly, if the transportation costs are again linear and

$$g_t + h_t^2 \leq h_t^1 + g_{t+1} \quad \text{for } t = 1, \dots, T-1$$

it is cheaper to transport as soon as we produce and store the production at the retailer. Then, without loss of optimality, we can assume that $I_t^1 = 0$ for all $t = 1, \dots, T$. These two special cases of (P) therefore yield a standard CLSP. In general, however, (P) does not reduce to a CLSP even if the transportation costs are linear.

In the remainder of this section we will assume without loss of generality that the inventory holding costs have been incorporated into the production and transportation cost functions, and can therefore be treated as zero and ignored.

4.4 Linear transportation costs

4.4.1 Introduction

In this section we will consider the case where the transportation costs are linear, i.e., $c_t(x) = g_t x$ for $t = 1, \dots, T$. We will develop a dynamic programming approach that finds the optimal costs of each subplan. As we will show, this algorithm runs in $O(T^2)$ time for a single subplan, but the costs of all $O(T^4)$ subplans can be computed simultaneously in $O(T^5)$ time. This algorithm constitutes Phase 1 of the dynamic programming approach. As mentioned above, Phase 2 of the approach can be carried out in $O(T^4)$ time, resulting in an $O(T^5)$ algorithm for solving this class of instances of (P).

4.4.2 Preprocessing

The cumulative demands of all sets of consecutive periods, denoted by $d_{t_1 t_2}$ can be determined in $O(T^2)$ time. In terms of the underlying network (as described in Section 2.2)

one unit produced in period t for satisfying demand in period $\tau \geq t$ will, in the optimal solution, flow along the minimum cost path from M_t to R_τ , assuming that transportation is allowed in all periods t, \dots, τ . This means that the minimal unit transportation costs associated with producing one unit in period t for consumption in period τ is equal to $G_{t\tau} \equiv \min(g_t, \dots, g_\tau)$, which can be computed for all relevant pairs (t, τ) a priori in $O(T^2)$ time. From Proposition 2.2 we know that production in any period will satisfy the entire demand in a consecutive set of periods, in addition to possibly part of the demands in the immediately preceding and following periods. Therefore, given the amount of cumulative production before period t , we know when the units produced in period t will be consumed. It will be useful to also determine, in a preprocessing step, the total transportation costs associated with producing, in period t , the entire demand of the consecutive periods $\tau_1 + 1, \dots, \tau_2$, assuming that transportation is allowed in all periods t, \dots, τ_2 , i.e.,

$$GD_{t\tau_1\tau_2} \equiv \sum_{r=\tau_1+1}^{\tau_2} d_r \min(g_t, \dots, g_r) = \sum_{r=\tau_1+1}^{\tau_2} d_r G_{tr}.$$

In $O(T^3)$ time, these costs can be calculated for all $t = 1, \dots, T$ and $t \leq \tau_1 + 1 \leq \tau_2 \leq T$.

This information will later enable us to compute the total transportation costs associated with production in period t in constant time. (Note that the running time of Phase 2 of the dynamic programming approach is $O(T^4)$, so that the preprocessing step does not impact the overall running time of the approach.)

4.4.3 The costs of a subplan

For a given subplan, we have seen in Section 4.2 that we can restrict our attention to solutions in which we produce to full capacity in exactly K periods. If the remaining demand ε is positive it will be produced in a single period. In the following we will assume that indeed $\varepsilon > 0$. It is straightforward to simplify the approach in case $\varepsilon = 0$.

Our approach will be to define, for each subplan $(t_1, t_2, \tau_1, \tau_2)$, a network with the property that $\phi(t_1, t_2, \tau_1, \tau_2)$ is equal to the shortest path between a pair of source and sink nodes in this network. All nodes in this network are of the form (t, y^c) , where t indicates a period, and y^c is equal to the cumulative production until period t . In particular, we define:

1. $(t_1, 0)$, which is the *source* node;
2. (t, y^c) for $t = t_1 + 1, \dots, t_2 - 1$ and $k \in \cup_{k=0}^K \{kb, kb + \varepsilon\}$, if $y^c \geq d_{\tau_1+1,t}$;
3. $(t_2, Kb + \varepsilon) \equiv (t_2, d_{\tau_1+1,\tau_2})$, which is the *sink* node.

Arcs in the network represent production decisions. Arcs are present between pairs of nodes in the network of the form (t, y^c) and $(t + 1, \bar{y}^c)$, when $\bar{y}^c - y^c \in \{0, \varepsilon, b\}$ if $y^c = kb$ for some integer k , and $\bar{y}^c - y^c \in \{0, b\}$ if $y^c = kb + \varepsilon$ for some integer k .

To illustrate this network, consider a problem instance with $b = 7$, and a subplan $(0, 3, 1, 5)$ with cumulative demand equal to 13. In this case, we will produce to full capacity in one period and produce $\varepsilon = 6$ units in another. Figure 4 illustrates the subplan, and gives the individual demands for the relevant periods. The structure of the

network that is used for computing the optimal subplan costs is illustrated in Figure 5. Note that node $(2, 0)$ has been disregarded since the demand in period 2 is larger than the cumulative production of this node, i.e., 0.

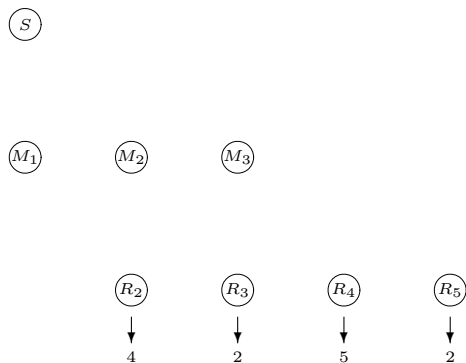


Figure 4: Subplan $(0, 3, 1, 5)$ with capacity $b = 7$.

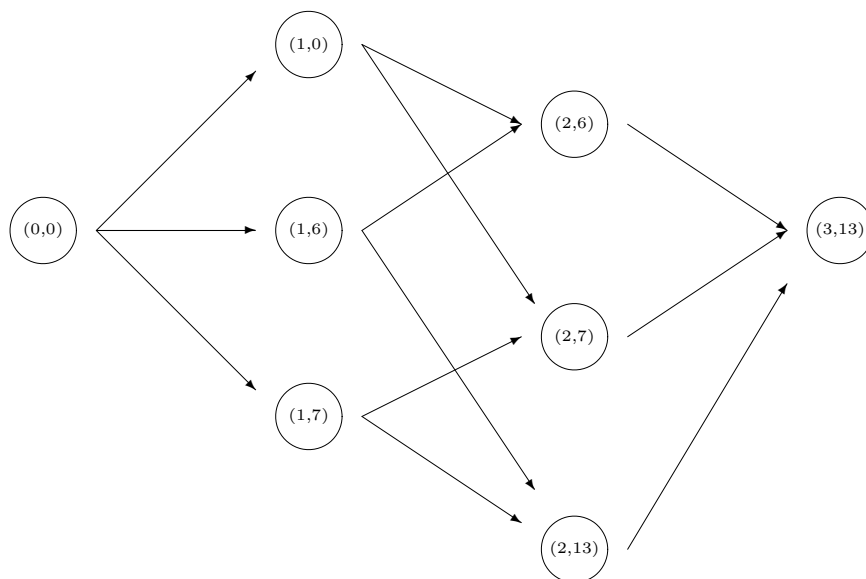


Figure 5: Structure of the network for finding $\phi(0, 3, 1, 5)$.

As mentioned above, each arc of the network described above represents a possible production decision. We let the costs of the arcs be equal to the total (production and transportation) costs associated with the production amount. Determining the production costs can be done in constant time given the preprocessing phase discussed in Section 4.2. As we will show below, the transportation costs can also be computed in constant time if, in addition to the information gathered in the preprocessing phase described in Section 4.4.2, we also find, for each node (t, y^c) in the network, the first period whose demand is not fully satisfied by the cumulative production y^c (say ℓ) as well as the part

of the demand of that period that remains to be satisfied (say δ). Using the cumulative demands $d_{\tau_1+1,t'}$ ($t' = \tau_1 + 1, \dots, \tau_2$) as well as the fact that the value of y^c can only be equal to kb or $kb + \varepsilon$ for $k = 0, \dots, K$, this additional information can be obtained in $O(T)$ time. As we will see later, this does not increase the running time of finding the costs of a single subplan.

Now consider an arc connecting the two nodes (t, y^c) (with first remaining demand period ℓ with remaining demand δ) and $(t + 1, \bar{y}^c)$ (with first remaining demand period $\bar{\ell}$ with remaining demand $\bar{\delta}$). When computing the corresponding transportation costs, we have to keep in mind that transportation can only take place in periods $\tau_1 + 1, \dots, t_2$. However, since the inventory holding costs are assumed to be equal to zero, this means that for purposes of computing the transportation costs we can assume that we are concerned with production that takes place in period $\max(t, \tau_1) + 1$. Similarly, we can assume that all demand in periods $t_2 + 1, \dots, \tau_2$ needs to be satisfied in period t_2 .

Distinguishing the various corresponding cases explicitly for the case where $\bar{y}^c - y^c \leq \delta$, the unit transportation costs of the quantity produced in period $t + 1$ are equal to

- $G_{t+1,\ell}$ if $t > \tau_1$ and $\ell \leq t_2$; or
- $G_{\tau_1+1,\ell}$ if $t \leq \tau_1$ and $\ell \leq t_2$; or
- G_{t+1,t_2} if $t > \tau_1$ and $\ell > t_2$; or
- G_{τ_1+1,t_2} if $t \leq \tau_1$ and $\ell > t_2$

which can be summarized by $G_{\max(t,\tau_1)+1,\min(\ell,t_2)}$. When $\bar{y}^c - y^c > \delta$, the transportation costs for this arc consist of up to three components:

- (i) the costs of transporting δ units, again at *unit* cost $G_{\max(t,\tau_1)+1,\min(\ell,t_2)}$;
- (ii) the costs of transporting $d_{\bar{\ell}} - \bar{\delta}$ units at *unit* cost $G_{\max(t,\tau_1)+1,\min(\bar{\ell},t_2)}$;
- (iii) the costs of transporting the remaining units produced, at *total* cost

$$GD_{\max(t,\tau_1)+1,\min(\ell,t_2),\min(\bar{\ell}-1,t_2)} + G_{\max(t,\tau_1)+1,t_2} \cdot d_{\max(\ell,t_2)+1,\max(\bar{\ell}-1,t_2)}$$

and can thus indeed be computed in constant time.

Any path in the network from the source $(t_1, 0)$ to the sink (t_2, d_{τ_1+1,t_2}) represents a feasible flow in the subplan $(t_1, t_2, \tau_1, \tau_2)$ with just one free production arc. Moreover, it is easy to see that the reverse is also true. Therefore, it suffices to find a minimal cost path in this network from the source node to the sink node. The time required for finding this minimal cost path is proportional to the number of arcs in the network. It is easy to see that both the number of nodes and the number of arcs in the network are $O(T^2)$, so that the cost of a single subplan can be determined in $O(T^2)$ time once the costs of all arcs are given.

Since there are $O(T^4)$ subplans, a straightforward application of the dynamic programming algorithm defined above to each individual subplan would yield an algorithm with running time $O(T^6)$ for computing the costs ϕ of all subplans. However, the running time can be reduced by observing that the costs of many subplans are related. In particular,

it is easy to see that the dynamic programming network corresponding to any subplan of the form $(t_1, t_2, \tau_1, \tau_2)$ is actually a subnetwork of the dynamic programming network for the subplan $(0, t_2, \tau_1, \tau_2)$. Therefore, solving for the shortest path between nodes $(0, 0)$ and $(t_2, d_{\tau_1+1, \tau_2})$ in the latter network yields, as a byproduct, the shortest paths between nodes $(t, 0)$ and (t_2, d_{τ_1+1, t_2}) for each $t = 1, \dots, \tau_1$. It thus follows that we essentially only need to consider the $O(T^3)$ subplans of the form $(0, t_2, \tau_1, \tau_2)$, the costs of which can be determined in $O(T^5)$ time. As an illustration, note that in Figure 5 the subnetwork rooted at node $(1, 0)$ is precisely the network corresponding to the subplan $(1, 3, 1, 5)$.

4.5 Fixed-charge transportation costs without speculative motives

4.5.1 Introduction

In this section we consider the case of fixed-charge transportation costs without speculative motives, i.e., $c_t(x) = f_t 1_{\{x>0\}} + g_t x$, where $1_{\{x>0\}}$ takes the value 1 if $x > 0$, and 0 otherwise, and $g_t \geq g_{t+1}$, for each $t = 1, \dots, T-1$. Similar to the analysis of the case of linear transportation costs in the previous section, we will determine the costs of each subplan using dynamic programming. This dynamic program runs in $O(T^3)$ time for each individual subplan, but we will show that the cost of all $O(T^4)$ subplans can be computed simultaneously in $O(T^6)$ time. Therefore, the running time of the dynamic programming approach to (P) with fixed-charge transportation costs without speculative motives is $O(T^6)$.

4.5.2 Zero inventory ordering property at the retailer

In this section we will show that, under fixed-charge transportation costs without speculative motives, solutions satisfying the ZIO property *at the retailer* are dominant. That is, given any feasible solution to the subplan $(t_1, t_2, \tau_1, \tau_2)$, there always exists another solution that is at least as good and satisfies the ZIO property at the retailer.

Theorem 4.1 *Given a subplan $(t_1, t_2, \tau_1, \tau_2)$, the set of solutions with the zero-inventory property at the retailer is dominant.*

Proof: Let $(\bar{y}, \bar{x}, \bar{I}^1, \bar{I}^2)$ be a feasible solution to the subplan $(t_1, t_2, \tau_1, \tau_2)$ that does not satisfy the ZIO property at the retailer. Let $\bar{t} \in \{\tau_1 + 1, \dots, t_2 - 1\}$ be a period so that

$$\bar{I}_{\bar{t}}^2 \bar{x}_{\bar{t}+1} > 0.$$

The positive inventory entering node $R_{\bar{t}}$ has been transported to the retailer level in some earlier period. However, due to the absence of speculative motives, the linear part of the transportation costs is non-increasing. We can thus reschedule the transportation of the $\bar{I}_{\bar{t}}^2$ units to period \bar{t} without increasing the costs. Repeating this argument for each period \bar{t} violating the zero-inventory property at the retailer, we obtain the desired result. \square

We may recall that $\phi(t_1, t_2, \tau_1, \tau_2)$ is equal to the minimal costs among the solutions of subplan $(t_1, t_2, \tau_1, \tau_2)$ with at most one free production arc. The following corollary to

Theorem 4.1 states that for finding this constrained minimum we can again restrict our search to solutions satisfying the ZIO property at the retailer.

Corollary 4.2 *The cost $\phi(t_1, t_2, \tau_1, \tau_2)$ associated with subplan $(t_1, t_2, \tau_1, \tau_2)$ can be found among all feasible solutions satisfying the ZIO property at the retailer.*

Proof: This follows immediately from the proof Theorem 4.1 by observing that the modification of the solution to obtain a solution satisfying the ZIO property does not alter the production flows. \square

An important consequence of Theorem 4.1 is that, if transportation takes place at some period t , the quantity transported is equal to the aggregate demand of a consecutive set of periods starting at t .

4.5.3 The costs of a single subplan

As in the case of linear transportation costs, we can restrict our attention to solutions for a subplan in which we produce to full capacity in exactly K periods (see Section 4.2). If the remaining demand $\varepsilon > 0$ it will be produced in a single period. In the following we will assume that indeed $\varepsilon > 0$. It is straightforward to simplify the approach in case $\varepsilon = 0$.

We will take a similar approach as for the linear cost case. However, due to the fixed-charge nature of the transportation cost functions, we need to explicitly keep track of which periods are transportation periods. This leads to a dynamic programming network for the computation of subplan costs that is an extension of the network used in Section 4.4.3, and where in addition the transportation costs are charged at the time of transportation, rather than at the time of production.

First note that each subplan $(t_1, t_2, \tau_1, \tau_2)$ can be easily reduced to a variant of the subplan (t_1, t_2, τ_1, t_2) where the demand of the retailer in period t_2 is replaced by $\sum_{t=t_2}^{\tau_2} d_t$. It therefore suffices to consider subplans of the form (t_1, t_2, τ_1, t_2) . We define an associated network which represents a forward recursion for calculating $\phi(t_1, t_2, \tau_1, t_2)$. All nodes in this network are of the form (t, y^c, s) , where t indicates a period, y^c is equal to the cumulative production until period t , and s represents the last period whose demand is satisfied using transportation up to period t . (If $s = t_2$, then no transportation will take place in this subplan after period t .) In particular, we define:

1. $(t_1, 0, \tau_1)$, which is the *source* node;
2. (t, y^c, s) for $t = t_1 + 1, \dots, t_2$, $y^c \in \cup_{k=0}^K \{kb, kb + \varepsilon\}$, and $s = \max(t, \tau_1), \dots, t_2$ if $y^c \geq d_{\tau_1+1, s}$.

We will call node $(t_2, Kb + \varepsilon, t_2) \equiv (t_2, d_{\tau_1+1, t_2}, t_2)$ the *sink* node.

Arcs in the network represent production and transportation decisions. Arcs are present between pairs of nodes in the network of the form (t, y^c, s) and $(t + 1, \bar{y}^c, \bar{s})$, when $\bar{y}^c - y^c \in \{0, \varepsilon, b\}$ if $y^c = kb$ for some integer k , $\bar{y}^c - y^c \in \{0, b\}$ if $y^c = kb + \varepsilon$ for some integer k , and $\bar{s} \geq s$. In addition, the ZIO property implies that if $s > t$, we can

restrict ourselves to arcs where $\bar{s} = s$, since no transportation will take place in period $t + 1$ (see also the discussion at the end of Section 3.1).

In Figure 6 we show the subplan $(0, 3, 1, 3)$, which is obtained from the subplan in Figure 4 by aggregating the demands in periods 3, 4, and 5. We illustrate the structure of the corresponding dynamic programming network described above in Figure 7. As before, the total demand is equal to 13 and the capacity is equal to 7, which means that we will produce to full capacity in one period and $\varepsilon = 6$ units in another. We may observe that all nodes associated with period 2 have transportation in that period. Moreover, due to the structure of the subplan, there are no nodes corresponding to transportation in period 1.

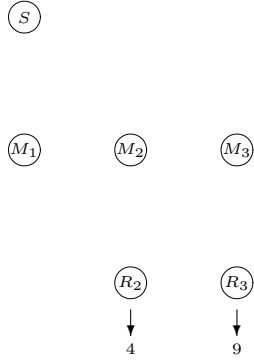


Figure 6: Subplan $(0, 3, 1, 3)$ and demands where $b = 7$.

The costs of an arc between nodes (t, y^c, s) and $(t + 1, \bar{y}^c, \bar{s})$ are given by

$$p_{t+1}(\bar{y}^c - y^c)$$

if $s > t$ (and therefore no transportation takes place in period $t + 1$), or

$$p_{t+1}(\bar{y}^c - y^c) + c_{t+1}(d_{t+1, \bar{s}})$$

(when period $t + 1$ is a transportation period, i.e., $s = t$). As in the linear cost case, determining the production costs can be done in constant time given the preprocessing phase discussed in Section 4.2. If we transport, the transportation costs of each individual arc in the network can also be computed in constant time if we, as we have done before, determine all aggregate demands of consecutive sets of periods $d_{tt'}$ in an overall preprocessing step, or even their corresponding transportation costs $c_t(d_{tt'})$. This preprocessing step takes $O(T^2)$ time.

It is clear that any path from the source node $(t_1, 0, \tau_1)$ to the sink node $(t_2, Kb + \varepsilon, t_2)$ represents a feasible solution to the subplan (t_1, t_2, τ_1, t_2) with the ZIO property at the retailer and at most one free production arc. Therefore, the problem of determining $\phi(t_1, t_2, \tau_1, t_2)$ reduces to finding the minimum cost path in the network from the source to the sink, which can be done in linear time in the number of arcs. It is easy to see that the number of nodes in the network is $O(T^3)$. More specifically, for each t there are $O(T)$ nodes of the form (t, \cdot, t) , and $O(T^2)$ nodes of the form (t, \cdot, s) with $s > t$. Each node of

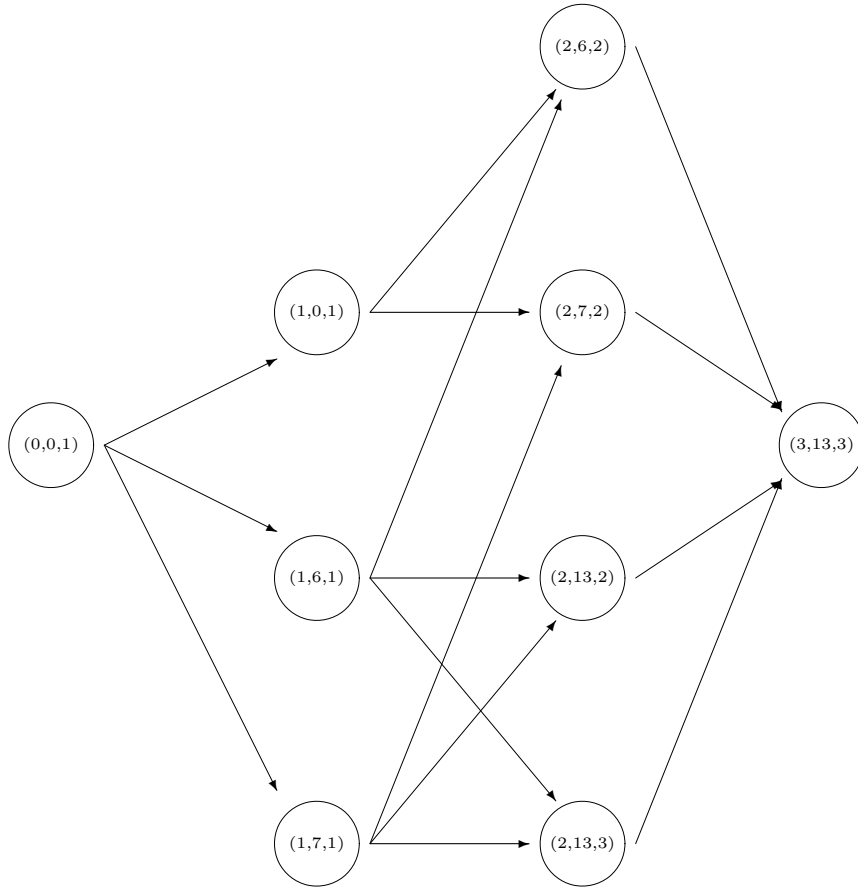


Figure 7: Structure of the network for finding $\phi(0, 3, 1, 3)$.

the form (t, \cdot, t) has $O(T)$ successors, and each node of the form (t, \cdot, s) with $s > t$ has $O(1)$ successors, which makes for a total of $O(T) \cdot (O(T) \cdot O(T) + O(T^2) \cdot O(1)) = O(T^3)$ arcs in the network. The total costs of a subplan of the form (t_1, t_2, τ_1, t_2) , and by the observation above actually of *any* subplan, can thus be determined in $O(T^3)$ time.

Since there are $O(T^4)$ subplans, a straightforward application of the dynamic programming algorithm defined above to each individual subplan would yield an algorithm with running time $O(T^7)$ for computing the costs ϕ of all subplans. However, as in the linear transportation cost case, the running time can be reduced by observing that the costs of many subplans are related. In particular, it is easy to see that the dynamic programming network corresponding to any subplan of the form (t_1, t_2, τ_1, t_2) is actually a subnetwork of the dynamic programming network for the subplan $(0, t_2, \tau_1, t_2)$. Therefore, solving for the shortest path between nodes $(0, 0, \tau_1)$ and $(t_2, d_{\tau_1+1, t_2}, t_2)$ in the latter network yields, as a byproduct, the shortest paths between nodes $(t, 0, \tau_1)$ and $(t_2, d_{\tau_1+1, t_2}, t_2)$ for each $t = 1, \dots, \tau_1$. It thus follows that we essentially only need to consider the $O(T^3)$ subplans of the form $(0, t_2, \tau_1, \tau_2)$, the costs of which can be determined in $O(T^6)$ time. As an illustration, note that in Figure 7 the subnetwork rooted at node $(1, 0, 1)$ is precisely the network corresponding to the subplan $(1, 3, 1, 3)$.

As a final remark, if the initial inventory at the manufacturer, I_0^1 , is strictly positive, the subplans with $t_1 = 0$ have a positive inventory inflow. If, in addition, $d_{\tau_1+1, \tau_2} < I_0^1$ the subplan is infeasible since any feasible flow in such a subplan will have $I_{\tau_2}^2 + I_{\tau_2}^1 > 0$, which violates the definition of a subplan. This case can thus easily be handled by, for all feasible subplans with $t_1 = 0$, simply computing K and ε as follows:

$$K = \left\lfloor \frac{d_{\tau_1+1, \tau_2} - I_0^1}{b} \right\rfloor$$

and

$$\varepsilon = d_{\tau_1+1, \tau_2} - I_0^1 - Kb.$$

For these subplans, we conclude that a node (t, y^c, s) should only be present if $I_0^1 + y^c \geq d_{1s}$. However, note that for subplans with $t_1 > 0$ we should have no positive inventory inflow. Therefore, in case $I_0^1 > 0$ we actually need to compute the costs of all subplans $(0, t_2, \tau_1, \tau_2)$ both with and without taking into account the initial inventory at the manufacturer. This clearly does not influence the running time of the algorithm.

4.6 Discussion

In this section we have proposed a two-phase approach to solving two-level lot-sizing problems with production capacities. The concavity of the objective function implies that there exists an extreme point optimal solution to the problem. Our algorithm uses this property, and is furthermore based on the result that any extreme point solution to the problem can be decomposed into a sequence of subplans, with the property that each subplan contains at most one production period in which production is strictly between 0 and capacity. In Phase 1 of the algorithm we compute the optimal costs of a subplan under this constraint, and given these costs, the dynamic programming formulation that is solved in Phase 2 of the algorithm captures all extreme point solutions to the lot-sizing problem.

A subplan consists of a set of production periods $t_1 + 1, \dots, t_2$, and a set of demand periods $\tau_1 + 1, \dots, \tau_2$. To ensure that a path from source to sink in the dynamic programming network in Phase 2 of the algorithm indeed corresponds to an extreme point solution, we need to restrict transportation within the subplan to the periods $\tau_1 + 1, \dots, t_2$. However, in the next section we will show that we may actually relax this restriction and allow transportation in the periods $t_1 + 1, \dots, \tau_2$ (while of course retaining the same production and demand periods). When we do this, a solution corresponding to a path in the Phase 2 dynamic programming network is not necessarily an extreme point solution to the lot-sizing problem anymore, since the conditions $I_{t_2}^1 = I_{\tau_1}^2 = 0$ may not be satisfied. However, it will be clear that any solution represented in the network is still a feasible solution to the lot-sizing problem. We will show that the length of a path in the network is never smaller than the cost of the corresponding solution, and in addition that the network contains an optimal solution to the lot-sizing problem with the property that its costs are equal to the corresponding path length. These properties imply that the Phase 2 dynamic programming with revised costs still solves the lot-sizing problem. For the two-level case, this seeming simplification of the algorithm does not translate into computational savings – the runtime of the heuristics for the two special cases remain $O(T^5)$ and $O(T^6)$, respectively. This is due to the fact that the time required for computing the costs of a subplan is independent of the form of the subplan. However, as we will see in the next section, in the multi-level case the revised dynamic program yields dramatic computational savings over a direct generalization of the original algorithm in which all extreme point solutions are represented.

5 The multi-level case

5.1 Introduction

We may extend the dynamic programming approach developed in Section 4.1 for the two-level case to the multi-level case, where again a Phase 2 dynamic programming network represents all extreme point solutions to the multi-level lot-sizing problem. To this end, we should define $\tilde{F}((\tau_\ell)_{\ell=1}^L)$ to be the minimum cost associated with satisfying the retailer demands in periods $\{\tau_L + 1, \dots, T\}$ using production in periods $\{\tau_1 + 1, \dots, T\}$, and warehouse ℓ in periods $\{\tau_\ell + 1, \dots, T\}$ for each $\ell = 1, \dots, L - 2$. We would then be interested in computing $F((0)_{\ell=1}^L)$. It is easy to see that the runtime of such an approach would be exponential in the number of levels, since the runtime of Phase 2 of the algorithm would be $O(T^{2L})$, which then would not include the time required to compute the costs, say $\phi((c_{\ell 1}, c_{\ell 2})_{\ell=1}^L)$, for all subplans. In the remainder of this section, we will formalize the discussion in Section 4.6, and show that we can actually solve the multi-level lot-sizing problem using the same dynamic programming approach as in the two-level case, where we replace the two-level subplan costs $\phi(t_1, t_2, \tau_1, \tau_2)$ by the minimum costs of satisfying demand in periods $\tau_1 + 1, \dots, \tau_2$ using production in periods $t_1 + 1, \dots, t_2$, where at most one of the production quantities may be different from both 0 and b , and where transportation at all levels is allowed in periods $t_1 + 1, \dots, \tau_2$. We will denote the latter costs by $\psi(t_1, t_2, \tau_1, \tau_2)$.

5.2 Formalization

It is easy to see that, as in the two-level case, each path from source to sink in the dynamic programming network corresponds to a *feasible* solution to the lot-sizing problem. However, the length of the path is not necessarily equal to the costs of the corresponding solution. The following theorem shows that the path length is never smaller than the actual costs of the solution, and is equal to the costs of the solution if all transportation and inventory cost functions are linear.

Theorem 5.1 *Each path in the Phase 2 dynamic programming network corresponds to a feasible solution to the lot-sizing problem. The length of this path cannot be smaller than the cost of the corresponding solution, and is equal to the solution cost if all transportation and inventory cost functions are linear.*

Proof: The fact that a path from the source to the sink in the Phase 2 dynamic programming network corresponds to a feasible solution to the lot-sizing problem follows immediately from the fact that all production capacity constraints as well as all demands are satisfied. However, certain transportation and inventory arcs may carry positive flow in the partial solutions corresponding to more than one arc in the path, and each of the partial flows is charged separately according to the corresponding cost function. Due to the concavity of all cost functions, it follows that the cost of the total flow will not exceed the sum of the costs of the individual flows on any particular arc, and therefore the length of a path will never be less than the costs of the corresponding solution. In addition, when all transportation and inventory cost functions are linear the path length and solution costs are clearly equal. \square

This theorem implies that the two-phase algorithm with revised arc costs solves the MCLSP with linear transportation and inventory holding costs. In the remainder of this section we will show that this result generalizes to the case with general concave transportation and inventory holding cost functions.

Lemma 5.2 *For any subplan $((s_{\ell 1}, s_{\ell 2})_{\ell=1}^L)$ we have that*

$$\phi((s_{\ell 1}, s_{\ell 2})_{\ell=1}^L) \geq \psi(s_{11}, s_{12}, s_{L1}, s_{L2}).$$

Proof: This result follows immediately by noting that both $\phi((s_{\ell 1}, s_{\ell 2})_{\ell=1}^L)$ and $\psi(s_{11}, s_{12}, s_{L1}, s_{L2})$ are the result of an optimization problem, where the feasible region of the former is a subset of the feasible region of the latter. \square

The next theorem shows that there exists an optimal solution to the lot-sizing problem that is represented by a path in the Phase 2 dynamic programming network whose length is equal to the optimal costs.

Theorem 5.3 *The Phase 2 dynamic programming network contains a path that corresponds to an optimal solution to our lot-sizing problem, and the length of the path is equal to the cost of this solution.*

Proof: Consider an extreme point optimal solution to the lot-sizing problem, say with cost Φ^* . As discussed in Section 2.3, this optimal solution decomposes into a sequence of subplans. It is easy to see that the Phase 2 dynamic programming network contains a path for which the production and demand periods of each of the arcs correspond to this sequence of subplans. Lemma 5.2 now says that the length of the path in the dynamic programming network, say Ψ , will not exceed Φ^* . However, by Theorem 5.1 we know that Ψ is an overestimation of the costs of a corresponding feasible solution. Optimality of Φ^* now implies that in fact $\Psi = \Phi^*$, which proves the desired result. \square

Theorems 5.1 and 5.3 clearly imply that the two-phase algorithm solves the MCLSP.

As in the two-level case, we can now conclude that Phase 2 of the algorithm runs in $O(T^4)$ time, given all values $\psi(t_1, t_2, \tau_1, \tau_2)$. In the remainder of this section, we will show that the entire algorithm can be implemented in polynomial time for the same cost structures as in the two-level case.

5.3 Cost functions

In a similar fashion as was done in Section 4.3, we may remove the inventory holding costs from the problem if the corresponding cost functions are linear, by a suitable redefinition of the production and transportation cost functions, while retaining concavity of the production cost functions, as well as the structure of the transportation functions (in the general concave, fixed-charge, as well as the linear case).

In Section 5.5, we assume that the transportation costs at any level have a fixed-charge structure without speculative motives. More formally,

$$c_t^\ell(x) = f_t^\ell 1_{\{x>0\}} + g_t^\ell x.$$

As in Section 4.3, the assumption that there are no speculative motives means that the variable inventory and transportation costs are such that it is attractive to transport as late as possible when given this choice. Equivalently, holding inventory at level ℓ is more attractive than holding inventory at level $\ell + 1$:

$$g_t^\ell + h_t^{\ell+1} \geq h_t^\ell + g_{t+1}^\ell \quad \text{for } t = 1, \dots, T-1; \ell = 1, \dots, L-1.$$

When reformulating the problem as a problem with zero inventory costs at all levels, the condition that there are no speculative motives implies that the (adjusted) variable transportation costs at each level are non-increasing.

As before, we will assume without loss of generality that the inventory holding costs have been incorporated into the production and transportation cost functions, and can therefore be treated as zero and ignored.

5.4 Linear transportation costs

In this section we will show how the methodology derived in Section 4.4 can be generalized to the multi-level case, to obtain an algorithm that computes all values of the function ψ in polynomial time, which in turn implies that the MCLSP is polynomially solvable

as well, under stationary capacities $b_t = b$ for $t = 1, \dots, T$, general concave production costs, linear inventory holding costs at all levels and linear transportation costs between all levels.

In a preprocessing stage, we can determine the minimal unit transportation costs associated with producing one unit in period t for consumption in period τ , which we will again call $G_{t\tau}$. All these values can be computed in $O(LT^2)$ time by solving T shortest path problems in acyclic graphs with $O(TL)$ arcs using backward recursion. Using these values, we can then again determine the total transportation costs associated with producing, in period t , the entire demand of the consecutive periods $\tau_1 + 1, \dots, \tau_2$, assuming that transportation is allowed in all periods t, \dots, τ_2 , i.e.,

$$GD_{t\tau_1\tau_2} \equiv \sum_{r=\tau_1+1}^{\tau_2} d_r G_{tr}.$$

In $O(T^3)$ time, these costs can be calculated for all $t = 1, \dots, T$ and $t \leq \tau_1 + 1 \leq \tau_2 \leq T$. We can now use exactly the same dynamic programming approach as in the two-level case to determine the values of ψ , where the costs of an arc in the corresponding network can now be determined much more easily than before, since we allow transportation in all periods $t_1 + 1, \dots, \tau_2$. This then yields an $O(T^5 + LT^2)$ algorithm for solving this variant of the MCLSP.

5.5 Fixed-charge transportation costs without speculative motives

In this section we will show how the methodology derived in Section 4.5 can be generalized to the multi-level case, to obtain an algorithm that computes all values of the function ψ in polynomial time. Therefore, the MCLSP is polynomially solvable under stationary capacities $b_t = b$ for $t = 1, \dots, T$, general concave production costs, linear inventory holding costs at all levels and fixed-charge transportation costs without speculative motives between all levels.

We first generalize Theorem 4.1 to the multi-level case.

Theorem 5.4 *Given a subplan $((\varsigma_{\ell 1}, \varsigma_{\ell 2})_{\ell=1}^L)$, the set of solutions with the zero-inventory property at all levels in $\{2, \dots, L\}$ is dominant.*

Proof: Let $(\bar{y}, \bar{x}, \bar{I})$ be a feasible solution to the subplan $((\varsigma_{\ell 1}, \varsigma_{\ell 2})_{\ell=1}^L)$ that does not satisfy the ZIO property at some level. Let $\bar{\ell}$ be the last level such that the ZIO property holds for all $\ell \in \{\bar{\ell} + 1, \dots, L\}$, but it is not true for level $\bar{\ell}$. We can construct a new solution at least as good as $(\bar{y}, \bar{x}, \bar{I})$ such that the ZIO property holds for all $\ell \in \{\bar{\ell}, \dots, L\}$. If $\bar{\ell} = 2$, then we have obtained the desired result. Otherwise, we repeat the procedure with the new solution. Observe that this procedure converges because the new $\bar{\ell}$ has decreased at least by one unit.

Let $\bar{t} \in \{\varsigma_{1\bar{\ell}} + 1, \dots, \varsigma_{2\bar{\ell}-1} - 1\}$ be a period so that

$$\bar{I}_{\bar{t}}^{\bar{\ell}} \bar{x}_{\bar{t}+1}^{\bar{\ell}-1} > 0.$$

The positive inventory $\bar{I}_{\bar{t}}$ has been transported to level $\bar{\ell}$ in some earlier period. However, due to the absence of speculative motives, the linear part of the transportation costs is non-increasing. In a similar fashion to Theorem 4.1, we can reschedule the transportation of the $\bar{I}_{\bar{t}}$ units to period \bar{t} without increasing the costs. Repeating this argument for each period \bar{t} violating the zero-inventory property at level $\bar{\ell}$, we obtain a solution where ZIO property is true for each level $\ell \in \{\bar{\ell}, \dots, L\}$. \square

Analogous to the two-level case, we have the following corollary.

Corollary 5.5 *The cost associated with subplan $((c_{\ell 1}, c_{\ell 2})_{\ell=1}^L)$ can be found among all feasible solutions satisfying the ZIO property at all levels in $\{2, \dots, L\}$.*

Proof: This follows immediately from the proof Theorem 5.4 by observing that the modification of the solution to obtain a solution satisfying the ZIO property does not alter the production flows. \square

This corollary implies that in any solution, any amount shipped is equal to the demand of a consecutive set of periods. Using this important property, we can use a slight modification of the dynamic programming algorithm that was used in the two-level case. Recall that the nodes in the corresponding network are of the form (t, y^c, s) . In the multi-level case, s represents the last period whose demand is satisfied using transportation from level 1 to level 2 up to period t . Arcs are now present between pairs of nodes in the network of the form (t, y^c, s) and (t, \bar{y}^c, \bar{s}) , when $\bar{y}^c - y^c \in \{0, \varepsilon, b\}$ if $y^c = kb$ for some integer k , $\bar{y}^c - y^c \in \{0, b\}$ if $y^c = kb + \varepsilon$ for some integer k , and $\bar{s} \geq s$. In the two-level case, we could use the ZIO property to further reduce the number of arcs in the network by observing that no transportation takes place in period $t + 1$ if $s > t$. However, in the multi-level case this is not necessarily true, as can be seen in the example in Figure 3.

The costs of an arc between nodes (t, y^c, s) and $(t + 1, \bar{y}^c, \bar{s})$ are now given by:

- the production costs: $p_{t+1}(\bar{y}^c - y^c)$;
- the transportation costs between levels 1 and 2: $c_{t+1}(d_{s+1, \bar{s}})$; and
- the transportation costs between levels 2 and L .

To determine the last cost category, we recall that Zangwill [18] uses the structure of extreme point solutions to the uncapacitated problem to derive a dynamic programming recursion in the quantities $C_{t\ell}(s_1, s_2)$, which are the optimal costs of shipping $d_{s_1 s_2}$ units from node (t, ℓ) to their destinations, i.e., demand nodes s_1, \dots, s_2 . We can thus use Zangwill's algorithm, in a preprocessing stage, to determine the values $C_{t2}(s_1, s_2)$ for all $t = 2, \dots, T$, $s_1 = t, \dots, T$, and $s_2 = s_1, \dots, T$ in $O(LT^4)$ time. The missing part of the arc costs are then equal to $C_{t+1,2}(s + 1, \bar{s})$. (Note that the quantities $C_{t1}(s_1, s_2)$ are not useful, since the inventory quantities at level 1 are not necessarily sums of demands of consecutive sets of periods.)

It is important to note that, although Zangwill's model allows for general concave transportation and inventory holding cost functions, we cannot use the same approach as described above in the presence of production capacities. The reason is that, in the

uncapacitated case, the ZIO property holds for arbitrary concave arc cost functions, while this is not the case in the capacitated case. However, as we have shown, in the case of linear inventory holding costs and fixed-charge transportation costs which exhibit no speculative motives we also obtain the ZIO property, enabling the use of Zangwill's algorithm to determine inputs to our algorithm.

It remains to determine the running time of our algorithm. Note that the structure of the dynamic programming network for computing the values of ψ is slightly different than in the two-level case due to the multi-level character of the problem. The network has the same nodes as in the two-level case, but $O(T^4)$ arcs in the multi-level case. (Note that this increase in running time is analogous to the increase in running time of Zangwill's algorithm for the uncapacitated case when we go from a two-level problem to more levels.) Using the same approach to computing multiple values of the function ψ at once, this yields an $O(T^7 + LT^4)$ algorithm for solving the multi-level variant of this problem.

6 Concluding remarks and future research

In this paper we have considered a generalization of the classical economic lot-sizing problem with stationary production capacities which allows for multiple levels of storage, as well as corresponding transportation decisions for transporting between the different levels. We have identified two important special cases of this problem that are solvable in polynomial time. The running times of the corresponding algorithms are remarkably insensitive to the number of levels in the supply chain.

Open issues for future research in this area can be divided into three directions. Firstly, the complexities, although polynomial in the planning horizon, are of relatively high order: $O(T^5)$ and $O(T^6)$, respectively, for the two-level cases, and slightly higher for the multi-level cases. It would be interesting if the order of the running time could be reduced, for instance by investigating whether more time can be saved by determining the costs of many or all subplans simultaneously. A second direction could be the generalization of the results to more general transportation cost functions, or, alternatively, a proof that the problem is NP-hard for other classes of transportation cost functions. Finally, it would be interesting to consider more complex supply chain structures, including, for example, product assembly structures at the producer level, or multiple retailers.

References

- [1] A. Aggarwal and J.K. Park. Improved algorithms for economic lot-size problems. *Operations Research*, 41(3):549–571, 1993.
- [2] R.K. Ahuja, T.L. Magnanti, and J.B. Orlin. *Network Flows: Theory, Algorithms, and Applications*. Prentice Hall, New Jersey, 1993.
- [3] K.R. Baker, P. Dixon, M.J. Magazine, and E.A. Silver. An algorithm for the dynamic lot-size problem with time-varying production capacity constraints. *Management Science*, 24(16):1710–1720, 1978.

- [4] G.R. Bitran and H.H. Yanasse. Computational complexity of the capacitated lot size problem. *Management Science*, 28(10):1174–1186, 1982.
- [5] P. Chandra and M.L. Fisher. Coordination of production and distribution planning. *European Journal of Operational Research*, 72:503–517, 1994.
- [6] C.-S. Chung and C.-H.M. Lin. An $O(T^2)$ algorithm for the $NI/G/NI/ND$ capacitated lot size problem. *Management Science*, 34(3):420–426, 1988.
- [7] J.R. Evans. An efficient implementation of the Wagner-Within algorithm for dynamic lot-sizing. *Journal of Operations Management*, 5:229–235, 1985.
- [8] A. Federgruen and M. Tzur. A simple forward algorithm to solve general dynamic lot sizing models with n periods in $O(n \log n)$ or $O(n)$. *Management Science*, 37:909–925, 1991.
- [9] M. Florian and M. Klein. Deterministic production planning with concave costs and capacity constraints. *Management Science*, 18:12–20, 1971.
- [10] M. Florian, J.K. Lenstra, and A.H.G. Rinnooy Kan. Deterministic production planning: algorithms and complexity. *Management Science*, 26(7):669–679, 1980.
- [11] A.M. Geoffrion and R.F. Powers. Twenty years of strategic distribution system design: an evolutionary perspective. *Interfaces*, 25(5):105–127, 1995.
- [12] C.P.M. van Hoesel and A.P.M. Wagelmans. An $O(T^3)$ algorithm for the economic lot-sizing problem with constant capacities. *Management Science*, 42(1):142–150, 1996.
- [13] F. Ortega. *Formulations and algorithms for fixed charge networks and lot-sizing problems*. PhD thesis, CORE, Louvain-la-Neuve, Belgium, 2001.
- [14] D.J. Thomas and P.M. Griffin. Coordinated supply chain management. *European Journal of Operational Research*, 94(1):1–15, 1996.
- [15] A. Wagelmans, S. van Hoesel, and A. Kolen. Economic lot sizing: An $O(n \log n)$ algorithm that runs in linear time in the Wagner-Within case. *Operations Research*, 40(1):S145–S156, 1992.
- [16] H.M. Wagner. A postscript to dynamic problems of the theory of the firm. *Naval Research Logistics Quarterly*, 7:7–12, 1960.
- [17] H.M. Wagner and T.M. Within. Dynamic version of the economic lot size model. *Management Science*, 5:89–96, 1958.
- [18] W.I. Zangwill. A backlogging model and a multi-echelon model of a dynamic economic lot size production system – a network approach. *Management Science*, 15(9):506–527, 1969.