Supply Chain Operations Planning with Setup Times and Multi Period Capacity Consumption

H. Zolfi [1], S.M.T. Fatemi Ghomi [1] and B. Karimi [1]
1 Department of Industrial Engineering
Amirkabir University of Technology, 424 Hafez Avenue, Tehran, Iran,
Tel.: +98 21 66413034; Fax +98 21 6641 3025
Home page: http://www.aut.ac.ir

Abstract

Developing an efficient heuristic algorithm to solve a *supply chain operations planning* model is the main purpose of this paper. The model considers multi period supply chain planning with capacitated resources. The concept of *multi period capacity consumption* has been developed recently at the context of supply chain management that realizes resource planning at a supply chain. Because of considering setup times and costs, the model contains binary variables. Since the mixed integer model is strongly NP-hard problem and finding a feasible solution is NP-complete, developing an efficient algorithm is remarkable. In this paper a heuristic algorithm is developed to solve this complicated model. Two reasons encouraged the authors to solve this complex problem. First, the model is an advanced and applicable operations planning model at the supply chain environment. Second, this model is strongly NP-hard. So it is of important task to develop a solution for the problem to be capable of feasible and efficient.

1 Introduction

De Kok and Fransoo (2003) define supply chain operations planning (SCOP) as coordinating material and resource release decisions in the supply chain such that predefined customer service levels are met at minimal cost. Extensive discussion about supply chain operations planning models can be found at De Kok and Fransoo (2003). A few researches in the past explained the concept of multi period capacity consumption. As a first work, Negnman (2000) described this concept as a new additional property for planned lead time. According to multi period capacity consumption, capacities of resources are allowed to consume at any internal periods during the lead time. A supply chain model with multi period capacity consumption and its concept can be comprehended clearly at Spitter et al. (2005). Their model did not consider the setup times and costs. So it was formulated as two LP[1] model. We develop our new model strongly based on the first model of Spitter et al. (2005). New model considers non zero setup times and costs. Considering non zero setup times and costs make it possible to be used at supply chains that setup times and costs are important. For example a part manufacturer factory can be considered as a good real example for our model. Consider that this factory is supplier of special parts as a member of Car Company's supply chain. It is producing some special parts for car manufacturing companies using a lot of bending, drilling and pressing machines. This factory must produce wide variety kinds of products. Since it is required to have none zero times and costs to changeover the setup of each machine, therefore setup times and costs are not negligible. It is apparent that all of machines are capacity restricted. At this situation the model of this paper is very suitable to determine lot sizes of each part on each machine. Additionally because of considering multi period capacity consumption, capacity planning of machines is more realistic and simultaneous planning of capacities and orders will be possible.

Models of supply chain planning problems and lot sizing models are very similar in many ways (see Voβ and Woodruff (2000)). So we can consider this new supply chain operations model as a new Multi Level Capacitated Lot Sizing Problem (MLCLSP) with setup times and costs. From this viewpoint the concept of multi period capacity consumption is considered at an MLCLSP with setup times. Multi period capacity consumption realizes the coordination of the release of materials and resources at the MLCLSP model.

From the aspect of solving the model, our model is complicated. Because of setup variables, the model will be an MIP[2] model (contrary to Spitter's model) and it is more complicated than the previous lot sizing models too. The most complex lot sizing model that has been considered yet is the MLCLSP with setup times with general assembly structure. This problem is NP-hard problem (Dellaert et al. (2000)) and finding a feasible solution is NP-complete (Maes et al. (1991)). As we know just two papers have discussed this model up to now. Tempelmeier and Derstroff (1996) solved this problem using Lagrange relaxation of capacity and inventory balance constraints and Katok et al. (1998) have solved the problem using LP based approach. The model of this paper will be more complicated considering the concept of multi period capacity consumption. Therefore developing a good feasible heuristic algorithm will be so important. The developed algorithm in this paper is based on decomposition the major model into two simpler models at a hierarchical structure and solving the model of each level efficiently. Also developed algorithm for these conditions can be used for models with zero setup times and costs. Therefore this model can be considered as an advanced and applicable supply chain planning model. Now the problem is how these operational situations can be modeled as a mathematical model and how the model can be solved that is strongly NP-hard.

[1] Linear Programming
[2] Mixed Integer Programming

2 The Problem formulation

To formulate the problem, it is required to define the notations which are very close to Spitter et al. (2005). *i, u, t* and *s* are indices of products, resources, order release times and resource release times respectively. *n, k* and *T* are number of items, resources and time horizon respectively. τ_i is the planned lead time of item *i*. h_{ij} indicates the number of units of item *i* used to produce one unit of item *j*. α_{it}, β_{it} and SC_{ius} are holding, backordering and setup costs respectively. The exogenously determined demand of item *i* for period *t* is shown by D_{it}. ST_{ius} is the setup time of item *i* on resource *u* at period *t*. C_{us} is the maximum available capacity of resource *u* at period *t*. R_{it} is the planned order of item *i* at time *t*, Z_{iuts} is the part of R_{it} produced on resource *u* at period *s* so that $s = t+1,\dots,T+\tau_i \mid s \le T$. Amount of R_{it} and Z_{iuts} are given for past periods ($t = -\tau_i + 1,\dots,-1$) as $\overline{R}_{it}$ and $\overline{Z}_{iuts}$. The binary setup variable is δ_{ius} that is 1 if setup of item *i* occurs on resource *u* at period *s* and 0 otherwise. Considering the concept of multi period capacity consumption and above notations, the problem is modeled as follows:

Problem A

$$Min \ \sum_{t=1}^{T}\sum_{i=1}^{n}(\alpha_{it} I_{it} + \beta_{it} B_{it}) + \sum_{u=1}^{k}\sum_{i=1}^{n}\sum_{s=1}^{T}\delta_{ius} SC_{ius} \tag{1}$$

$s.t.$:

$$R_{it-\tau_i} + I_{it-1} - I_{it} - D_{it} - \sum_{j=1}^{n} h_{ij} R_{jt} + B_{it} - B_{it-1} = 0 \qquad \forall\, t = 1,\dots,T,\ i = 1,\dots,n \tag{2}$$

$$R_{it} = \sum_{\substack{s=t+1 \\ s<=T}}^{t+\tau_i} \sum_{u=1}^{k} Z_{iuts} \qquad \forall\, i = 1,\dots,n \quad ,t = -\tau_i + 1,\dots,T-1 \tag{3}$$

$$\sum_{i=1}^{n} \sum_{t=s-\tau_i}^{s-1} Z_{iuts} + \sum_{i=1}^{n} \delta_{ius} ST_{ius} \le C_{us} \qquad \forall\, u = 1,\dots,k \quad ,s = 1,\dots,T \tag{4}$$

$$\delta_{ius} \ge \frac{\sum_{t=s-\tau_i}^{s-1} Z_{iuts}}{M(=C_{us})} \qquad \forall\, i = 1,\dots,n \quad ,u = 1,\dots,k \quad ,s = 1,\dots,T \tag{5}$$

$$B_{it} - B_{it-1} \le D_{it} \qquad \forall\, i = 1,\dots,n \quad ,t = 1,\dots,T \tag{6}$$

$$R_{it} = \overline{R}_{it} \qquad \forall\, i = 1,\dots,n \quad ,t = -\tau_i + 1,\dots,-1 \tag{7}$$

$$Z_{iuts} = \overline{Z}_{iuts} \qquad \forall\, i = 1,\dots,n \quad ,u = 1,\dots,k \quad ,t = -\tau_i + 1,\dots,-1 \quad ,s = t+1,\dots,0 \tag{8}$$

$$R_{it}, B_{it}, I_{it} \ge 0 \qquad \forall\, i = 1,\dots,n \quad ,t = 1,\dots,T \tag{9}$$

$$Z_{iuts} \ge 0 \qquad i = 1,\dots,n \quad ,u = 1,\dots,k \quad ,t = -\tau_i + 1,\dots,T-1 \quad ,s = t+1,\dots,t+\tau_i \mid s \le T \tag{10}$$

$$\delta_{ius} \in \{0,1\} \qquad \forall\, i = 1,\dots,n \ ,u = 1,\dots,k \ ,s = 1,\dots,T \tag{11}$$

The objective function minimizes the total inventory holding, backorder and setup costs. Constraint set (2) implies the inventory balance equations. Equation (3) presents the relation between orders and their feasible productions according to multi period capacity consumption. Constraint set (4) presents capacity feasibility on each resource and at each period. The constraints set (5) mean if there is at least one positive Z_{iuts} ($t \in [s-\tau_i, s-1]$), setup of item *i* occurs on resource u at time slot s. C_{us} is set as an upper bound for $\sum_{t=s-\tau_i}^{s-1} Z_{iuts}$. Equation (7) enforces that backorders are allowed only for end items and amount of backordering is less than the independent exogenous demand. At equations (7) and (8) the effect of past orders and productions which still have influence on the future is considered. Equations (9) and (10) are sign restrictions and equation (11) claims δ_{ius} is binary variable.

3 The heuristic algorithm

The above MIP model is very hard to solve. The major factors of complexity are:

1) Relation between orders at different levels in the inventory balance equations.
2) Capacity constraints.
3) Existence of setup times in the capacity constraints.
4) Different summation bounds on the production variables in the various equations.

We tackle the complexity of this problem by hierarchical planning. The main model is decomposed into two new related models. At the first level we try to overcome the first factor of complexity. Therefore at the second level there will be a **single level**, multi item, multi period and multi resource problem. By Lagrange relaxing the equation (3) at the model of second level, several single item, single resource and single period MIP models are obtained and thus remaining factors of complexity are controlled. In this way the first model is engaged to plan orders, based on the exact holding and backordering costs and approximate setup costs. At the second level, planed orders are split up to the lot sizes; based on setup times and costs. Considering the cited points, problem A is disintegrated as follows:

Problem B1: a linear model for the first level of hierarchy

$$Min \quad \sum_{t=1}^{T}\sum_{i=1}^{n}(\alpha_{it} I_{it} + \beta_{it} B_{it}) + \sum_{u=1}^{k}\sum_{i=1}^{n}\sum_{s=1}^{T} lc_{ius} SC_{ius} \tag{12}$$

s.t:

$$R_{it-\tau_i} + I_{it-1} - I_{it} - D_{it} - \sum_{j=1}^{n} h_{ij} R_{jt} + B_{it} - B_{it-1} = 0 \qquad \forall\, t = 1,\dots,T \quad, i = 1,\dots,n \tag{13}$$

$$R_{it} = \sum_{\substack{s=t+1 \\ s<=T}}^{t+\tau_i} \sum_{u=1}^{k} Z_{iuts} \qquad \forall\, i = 1,\dots,n \quad, t = -\tau_i + 1,\dots,T-1 \tag{14}$$

$$\sum_{t=s-\tau_i}^{s-1} Z_{iuts} \le V_{ius} \qquad \forall\, i = 1,..,n \quad, u = 1,\dots,k \quad, s = 1,\dots,T \tag{15}$$

$$\sum_{i=1}^{n} V_{ius} \le C_{us} \qquad \forall\, u = 1,\dots,k \quad, s = 1,\dots,T \tag{16}$$

$$B_{it} - B_{it-1} \le D_{it} \qquad \forall\, i = 1,\dots,n \quad ,t = 1,\dots,T \tag{17}$$

$$R_{it} = \overline{R}_{it} \qquad \forall\, i = 1,\dots,n \quad ,t = -\tau_i + 1,\dots,-1 \tag{18}$$

$$Z_{iuts} = \overline{Z}_{iuts} \qquad \forall\, i = 1,\dots,n \quad ,u = 1,\dots,k \quad ,t = \tau + 1,\dots,-1 \quad ,s = t + 1,\dots,0 \tag{19}$$

$$R_{it}, B_{it}, I_{it} \ge 0 \qquad \forall\, i = 1,\dots,n \quad ,t = 1,\dots,T \tag{20}$$

$$Z_{iuts} \ge 0 \qquad i = 1,\dots,n \quad ,u = 1,\dots,k \quad ,t = -\tau_i + 1,\dots,T - 1 \quad ,s = t + 1,\dots,t + \tau_i \mid s \le T \tag{21}$$

$$V_{ius} \ge 0 \qquad \forall\, i = 1,..,n \quad ,u = 1,\dots,k \quad ,s = 1,\dots,T \tag{22}$$

Problem B2: a single level MIP model for the second level of hierarchy

$$Min \quad \sum_{u=1}^{k}\sum_{i=1}^{n}\sum_{s=1}^{T} \delta_{ius}\, SC_{ius} \tag{23}$$

$$R_{it} = \sum_{\substack{s=t+1 \\ s<=T}}^{t+\tau_i}\sum_{u=1}^{k} \mathrm{Z}_{\mathrm{iuts}} \qquad \forall\, i = 1,\dots,n \quad ,t = -\tau_i + 1,\dots,T - 1 \tag{24}$$

$$\sum_{t=s-\tau_i}^{s-1} Z_{iuts} + \delta_{ius}\, ST_{ius} \le V_{ius} \qquad \forall i = 1,\dots n \quad ,u = 1,\dots,k \quad ,s = 1,\dots,T \tag{25}$$

$$\delta_{ius} \ge \frac{\sum_{t=s-\tau_i}^{s-1} Z_{iuts}}{V_{ius} - ST_{ius}} \qquad \forall\, i = 1,\dots,n \quad ,u = 1,\dots,k \quad ,s = 1,\dots,T \tag{26}$$

$$Z_{iuts} = \overline{Z}_{iuts} \qquad \forall\, i = 1,\dots,n \quad ,u = 1,\dots,k \quad ,t = \tau + 1,\dots,-1 \quad ,s = t + 1,\dots,0 \tag{27}$$

$$Z_{iuts} \ge 0 \qquad i = 1,\dots,n \quad ,u = 1,\dots,k \quad ,t = -\tau_i + 1,\dots,T - 1 \quad ,s = t + 1,\dots,t + \tau_i \mid s \le T \tag{28}$$

$$\delta_{ius} \in \{0,1\} \qquad \forall\, i = 1,\dots,n \quad ,u = 1,\dots,k \quad ,s = 1,\dots,T \tag{29}$$

Simultaneous consideration of constraints (15) and (16) implies constraint (4) without setup times. Setup times are studied at the problem B2 exactly. At the objective function of problem B1, δ_{ius} is replaced with linear estimator called lc_{ius}. Based on *fixed charge problems* literature (e.g. see Taha (1975)), the definition of lc_{ius} can be stated as follows:

$$lc^{q+1}{}_{ius} = \begin{cases} \dfrac{\sum_{t=s-\tau_i}^{s-1} Z^{q+1}{}_{iuts}}{\sum_{t=s-\tau_i}^{s-1} Z^{q}{}_{iuts}} & if \ \sum_{t=s-\tau_i}^{s-1} Z^{q}{}_{iuts} > 0 \\ 0 & if \ \sum_{t=s-\tau_i}^{s-1} Z^{q}{}_{iuts} = 0 \end{cases} \tag{30}$$

lc_{ius} behaves as δ_{ius} implicitly. δ_{ius} is replaced with lc_{ius}, because it is desired to have an LP problem at the first level. lc_{ius} is determined in the iterative way. At iteration q+1 ($q \ge 0$), Z^{q+1}_{iuts} is the variables of current iteration and Z^{q}_{iuts} is the solution of problem B1 at the previous iteration. After problem B1 is solved the initial solution (not necessarily feasible for problem A) is gained and initial values for I_{it}, B_{it}, R_{it}, Z_{iuts} and V_{ius} are obtained. If no feasible solution exists for problem B1, the problem A is infeasible. Otherwise at problem B2 we try to find a feasible good solution restricted to obtained R_{it}. Fixing R_{it} at problem B2 implies fixed I_{it} and B_{it}, and fixing these variables at the first level makes the second level independent from inventory balance equations. At the next step the algorithm tries to find a feasible solution.

- Consider $\sum_{t=s-\tau_i}^{s-1} Z_{iuts}$ and define δ_{ius} according to the following equation:

$$\begin{cases} if \quad \sum_{t=s-\tau_i}^{s-1} Z_{iuts} > 0 \quad \Rightarrow \quad \delta_{ius} = 1 \\ if \quad \sum_{t=s-\tau_i}^{s-1} Z_{iuts} = 0 \quad \Rightarrow \quad \delta_{ius} = 0 \end{cases} \tag{31}$$

- Correct V_{ius} to obtain feasible equation (25) as follows:

$$\begin{cases} if \quad \sum_{t=s-\tau_i}^{s-1} Z_{iuts} > 0 \quad \Rightarrow \quad V_{ius} = \sum_{t=s-\tau_i}^{s-1} Z_{iuts} + ST_{ius} \\ if \quad \sum_{t=s-\tau_i}^{s-1} Z_{iuts} = 0 \quad \Rightarrow \quad V_{ius} = 0 \end{cases} \tag{32}$$

By changing V_{ius} as above, problem B2 becomes feasible and all constraints of problem B1 remain feasible except constraint (16) that may violate feasibility. In this step the algorithm tries to satisfy constraint (16) through decreasing V_{ius} so that other constraints remain feasible. Decreasing V_{ius} may just break up the feasibility of constraint (25). To prevent Z_{iuts} infeasibility of constraint (25), if V_{ius} decreases for special *i*, *u* and *s*, some related Z_{iuts} are shifted to other periods or on the other resources that called *the destination point of transition* and indicated by "*". The indices of selected Z_{iuts} to shift is called as *the origin point of transition* and signed by "$^{-}$". The destination point is selected according to some conditions such that R_{it} is not allowed to change. Following pseudo code consists of detailed description of this step:

$for\ \bar{s} = 1\ to\ T$

$for\ \bar{u} = 1\ to\ k$

$if\ constraint\ (16)\ is\ infeasible$

$select\ the\ smallest\ positive\ \sum_{t=\bar{s}-\tau_i}^{\bar{s}-1} Z_{i\bar{u}t\bar{s}}$ and indicate related i with $\bar{\mathrm{i}}$

$\text{for } \bar{\mathrm{t}} = \bar{s} - \tau_{\bar{i}}\ \ to\ \ \bar{s} - 1$

$\text{if } Z_{\bar{i}\bar{u}\bar{t}\bar{s}} > 0$

$\text{for } u^* = 1\ \ to\ \ k$

$\text{for } \mathrm{s}^* = \bar{\mathrm{t}} + 1\ \ to\ \ \min(\mathrm{T}, \bar{\mathrm{t}} + \tau_{\bar{\mathrm{i}}})$

$if\ C_{u^*s^*} - \sum_{i=1}^{n} V_{iu^*s^*} > 0$

$if\ \ \sum_{t=s^*-\tau_{i^*}}^{s^*-1} Z_{i^*u^*ts^*} > 0$

$$\begin{cases} i^* = \bar{i}\ , \quad \mathrm{t}^* = \bar{t} \\ transfer = \min(C_{u^*s^*} - \sum_{i=1}^{n} V_{iu^*s^*}\ ,\ Z_{\bar{i}\bar{u}\bar{t}\bar{s}}) \\ Z_{i^*u^*t^*s^*} = Z_{i^*u^*t^*s^*} + transfer \\ Z_{\bar{i}\bar{u}\bar{t}\bar{s}} = Z_{\bar{i}\bar{u}\bar{t}\bar{s}} - transfer \\ V_{i^*u^*s^*} = V_{i^*u^*s^*} + transfer \\ V_{\bar{i}\bar{u}\bar{s}} = V_{\bar{i}\bar{u}\bar{s}} - transfer \end{cases}$$

After a feasible solution is found at previous step, the algorithm is going to find a good near optimal solution for problem B2. To solve problem B2, first it is decomposed using Lagrange relaxation of constraints (24). Lagrange multipliers are defined as follows:

λ_{it} : Lagrange multiplier for constraint $R_{it} = \sum_{\substack{s=t+1 \\ 1\le s\le T}}^{t+\tau_i} \sum_{u=1}^{k} Z_{iuts}$, $i=1,\dots,n$, $t=-\tau_i+1,\dots,T-1$

the relaxed problem of B2 after rearranging of objective function is:

Problem RB2:

$$\min \quad \sum_{i=1}^{n}\sum_{u=1}^{k}\sum_{s=1}^{T}\delta_{ius}SC_{ius} - \sum_{i=1}^{n}\sum_{u=1}^{k}\sum_{s=1}^{T}\sum_{t=s-\tau_i}^{s-1}\lambda_{it} Z_{iuts} \tag{34}$$

$s.t.$:

$$\sum_{t=s-\tau_i}^{s-1} Z_{iuts} + \delta_{ius} ST_{ius} \le V_{ius} \qquad \forall i=1,\dots n \quad ,u=1,\dots,k \quad ,s=1,\dots,T \tag{35}$$

$$\delta_{ius} \ge \frac{\sum_{t=s-\tau_i}^{s-1} Z_{iuts}}{V_{ius}-ST_{ius}} \qquad \forall\ i=1,\dots,n \quad ,u=1,\dots,k \quad ,s=1,\dots,T \tag{36}$$

$$Z_{iuts} \ge 0 \qquad i=1,\dots,n \quad ,u=1,\dots,k \quad ,s=1,\dots,T \quad ,t=s-\tau_i,\dots,s-1 \tag{37}$$

$$\delta_{ius} \in \{0,1\} \qquad \forall\ i=1,\dots,n\ ,\ u=1,\dots,k,\ s=1,\dots,T \tag{38}$$

The problem RB2 can be decomposed into *nkT* single item, single resource and single period problems. Each of decomposed problems is solved using the following subroutine:

1) *if* $V-ST \le 0$ *then* $Z_t = 0 \quad \forall t \in [s-\tau_i, s-1]$ *and* $\delta = 0$.
2) *if* $V-ST > 0$ *and if* $\lambda_t \le 0\,(\forall t \in [s-\tau_i, s-1])$ *then* $Z_t = 0 \quad \forall t \in [s-\tau_i, s-1]$ *and* $\delta = 0$
3) *if* $V-ST > 0$ *and if there is at least one* $\lambda_t > 0$
4) *select the maximum* $\lambda_t > 0$ *(denoted by* λ_{t^*}*)*
5) *if* $SC - \lambda_{t^*}(V-ST) < 0$ *then* $(Z_{t^*} = V-ST$ *and* $\delta = 1$ *and* $Z_t = 0\ (\forall t \in [s-\tau_i, s-1] \ne t^*))$
6) *else* $Z_t = 0 \quad \forall t \in [s-\tau_i, s-1]$ *and* $\delta = 0$

Problem B2 is solved iteratively by updating Lagrange multipliers at the each iteration. We use subgradiant optimization technique to update Lagrange multipliers. Therefore, at each iteration problem RB2 must be solved to obtain new lower and upper bounds for problem B2. The objective function value of new optimal solution for problem RB2 is a new lower bound for problem B2. At the each iteration of subgradiant optimization method, the best lower bound of problem B2 is updated by selecting the maximum lower bounds obtained up to now.

4 Our Results

The algorithm has been tested using randomly generated problems. The quality of solutions of the heuristic algorithm is evaluated using LINGO optimal solutions for the small size examples. For small size examples (average binary variables is 50), the solutions obtained by heuristic algorithm are, on average, 2.69% worse than the optimal solution. Katuk et al. (1998) have reported average 4% of optimality gap for their algorithm. Comparing with their solutions, our algorithm performs better. For all small examples, the heuristic algorithm could find the feasible solution and, on average, about 25% of solutions were optimum. Running time of the small size problems is not reported because it is negligible (less than 2 seconds).

LINGO can not find the optimum solutions of the medium size (average binary variables is 500) and the large scale examples (average binary variables is 1500) at reasonable time. Therefore, we evaluate the heuristic algorithm using time-truncated LINGO solutions (close to Katok et al. (1998)). For medium size examples the solutions of LINGO, on average, are 3.22% better than the heuristic solutions that are acceptable with respect to the complexity of the problem. For large scale examples at 64.12% of the problems, LINGO could not find a feasible solution at runtime of the heuristic. According to the average deviation, it is clear that the heuristic is very efficient comparing to LINGO solutions, because the heuristic have produced the solutions with -317.18% deviation from LINGO. The number of problems that the heuristic solutions are better than the LINGO solutions is 94.87% that proves high performance of the heuristic. The magnitude of 16.73% is obtained for $\frac{upper\ bound - lower\ bouned}{lower\ bouned}\%$ that is comparable with 16.5% reported by Tempelmeier and Derstroff (1996). The results indicated a good performance for small size and medium size problems, and high performance for large scale problems.

5 References

Dellaert, N., Jeunet, J., Jonard, N., 2000. A genetic algorithm to solve the general multi-level lot-sizing problem with time varying costs. International Journal of Production Economics 68, 241–257.

De Kok, A. G., Fransoo, J. C., 2003. Planning supply chain operations: Definition and comparison of planning concepts. Pages 597–675 of: De Kok, A. G., & Graves, S. C., (eds), Handbook in Operations Research and Management Science, Volume 11: Design and Analysis of Supply Chains. Amsterdam.

Katok, E., Lewis, H.S., Harrison, T.P., 1998. Lot sizing in general assembly systems with setup costs, setup times, and multiple constrained resources. Management Science 44 (6), 859–877.

Maes, J., McClain, J.O., Van Wassenhove, L.N., 1991. Multilevel capacitated lot sizing complexity and LP-based heuristics. European Journal of Operational Research 53, 131–148.

Negenman, E. G., 2000. Material coordination under capacity constraints. Ph.D. thesis, Eindhoven University of Technology, Eindhoven, Netherlands.

Spitter, J. M., Hurkens, C. A. J., de Kok, A. G., Lenstra, J. K., Negenman, E. G., 2005. Linear programming models with planned lead times for supply chain operations planning. European Journal of Operational Research 163, 706–720.

Taha, H. A., 1975. Integer Programming, Theory, Applications and Computations. Academic Press Inc., Florida, USA.

Tempelmeier, H., Derstroff, M., 1996. A Lagrangean-based heuristic for dynamic multi-item multi-level constrained lot sizing with setup times. Management Science 42, 738–757.

Voß, S., Woodruff, D. L., 2003. Introduction to Computational Optimization Models for Production Planning in a Supply Chain. Springer, Berlin.