SOLVING THE MULTI-LEVEL CAPACITATED LOT SIZING PROBLEM VIA DUAL REOPTIMIZATION

Florian Sahling
Stefan Helber

Universität Hannover
Wirtschaftswissenschaftliche Fakultät
Lehrstuhl für Produktionswirtschaft
Tel. ++49 511 762 5650
{*florian.sahling, stefan.helber*}*@prod.uni-hannover.de*

Abstract: We consider the problem to determine lot sizes for a multi-level production system producing discrete products. The time-varying demand per period is assumed to be given and has to be satisfied. The production facilities at the different production stages are capacity constrained. The objective is to find a feasible production plan with minimal setup and holding costs. The numerical procedure used to solve the problem is based on tabu search to determine setup patterns and on dual re-optimization to compute lot sizes.

Keywords: Capacitated lot sizing, MLCLSP

1 Introduction

We consider a production system where multiple machines or other resources with limited capacity are used to produce several end products. Consider the example in Figure 1.

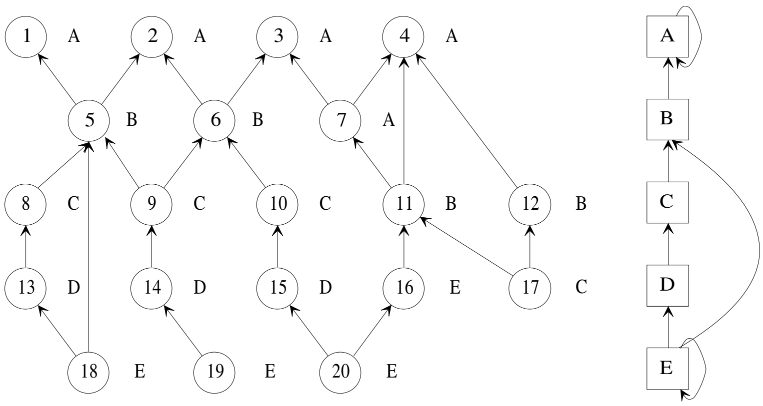

Figure 1: Product and Resource Structure

The bill of material and the process plans for the different parts have been used to determine the necessary operations to the four end products 1 to 4. Throughout this paper, we follow the convention to refer to the result of a *single* operation as a product and assume that for each operation exactly one machine is required. The structure in the left part of Figure 1 is hence called the “product structure”. In the given example, we assume that five different machines A to E are used in the production process. The precedence relations between the machines are given in the so-called “resource structure”. Note that while the product structure is non-cyclic as no operation can be its own predecessor, the resource structure may in general be cyclic.

Table 1: Notation for MLCLSP

Parameters:	
a_{ki}	number of units of product k required to produce one unit of product i
b_{jt}	available capacity of resource j in period t
B	big number
d_{kt}	external demand of product k in period t
h_k	holding cost of product k per unit and period
J	number of resources $(j = 1, \dots, J)$
K	number of products $(k = 1, \dots, K)$
$\mathcal{K}_j$	set of products requiring resource j
$\mathcal{N}_k$	set of immediate successors of product k
s_k	setup cost of product k
T	number of periods
tp_k	production time per unit of product k
ts_k	setup time of product k
z_k	planned lead time of product k
Decision variables:	
q_{kt}	production quantity (lot size) of product k in period t
y_{kt}	planned end-of-period inventory of product k in period t
γ_{kt}	binary setup variable of product k in period t

We assume that the planning horizon is divided into time periods and that for each period and product the demand is given and has to be satisfied. Note that the external demand is not restricted to the end products. There may also be external demand for intermediate products which are used as spare parts.

Whenever a product is produced during a time period, a setup is required which results in both setup costs and setup times. The setup state of any resource is assumed to be lost at the end of a period. Given the limited capacity of the machines and the setup costs, it may be necessary or economically attractive to produce some units of a product earlier than necessary with respect to its demand. In this case, a planned inventory is build up which results in holding costs.

Note that we do not aim at a detailed production schedule for the machines as we do not take the sequencing of the different products within a given period into account. In the presence of multiple production stages, it can be impossible to derive a detailed schedule from the lot-sizing decision unless we allow for a positive lead time.

While there is a lot of published research on dynamic lot sizing, the literature on this so-called Multi-Level Capacitated Lot Sizing Problem (MLCLSP) is rather limited, see [1, 4, 5, 10, 11, 12] for some earlier work. More recent results are presented in [6], see also the book by Tempelmeier [9] for a current overview.

The remainder of the paper is structured as follows. In Section 2 we give a mathematically precise formulation of the optimization problem. Based on this original formulation, some additional models are derived which motivate a new algorithm to solve the MLCLSP. The algorithm is outlined in Section 4. In this algorithm, the problem is split up into a decision about the setup pattern and a second decision about the production quantities for a given setup pattern. The latter problem is solved via the iterative re-optimization of the dual version of an extended model as proposed by Meyr and Fleischmann [3, 7, 8] for the General Lot Sizing and Scheduling Problem (GLSP). This is a promising approach as it relies to a large extent on standard procedures and software to solve parts of the problem. As the procedure is not very problem-specific, we are confident that it can be adapted to modifications of the MLCLSP. Preliminary numerical results are presented in Section 4 and some directions for future research are given in Section 5.

2 The Multi-Level Capacitated Lot-Sizing Problem (MLCLSP)

Based on the problem description given in Section 1, we now formally state the MLCLSP using the notation in Table 1.

Model MLCLSP

$$\text{Minimize } Z = \sum_{k=1}^{K} \sum_{t=1}^{T} (s_k \cdot \gamma_{kt} + h_k \cdot y_{kt}) \tag{1}$$

subject to:

$$y_{k,t-1} + q_{k,t-z_k} - \sum_{i \in \mathcal{N}_k} a_{ki} \cdot q_{it} - y_{kt} = d_{kt} \qquad \forall\, k, t \tag{2}$$

$$\sum_{k \in K_j} (tp_k \cdot q_{kt} + ts_k \cdot \gamma_{kt}) \leq b_{jt} \qquad \forall\, j, t \tag{3}$$

$$q_{kt} - B \cdot \gamma_{kt} \leq 0 \qquad \forall\, k, t \tag{4}$$

$$q_{kt} \geq 0 \qquad \forall\, k, t \tag{5}$$

$$y_{kt} \geq 0 \qquad \forall\, k, t \tag{6}$$

$$\gamma_{kt} \in \{0, 1\} \qquad \forall\, k, t \tag{7}$$

$$y_{k0} = y_{kT} = 0 \qquad \forall k \tag{8}$$

The objective function (1) states that the sum of the setup and holding costs has to be minimized. The inventory balance equation (2) reflects the multi-level production structure. Note that due to the lead time z_k, a production quantity of product k in period $t - z_k$ is available in period t to satisfy external demand d_{kt} or to be used in the production of the succeeding product i. If the time periods are rather long, it may be possible to set $z_k = 0, \forall k$ and still derive feasible detailed production schedules. If we set $z_k = 1, \forall k$, any feasible lot-sizing decision can always be transformed into a feasible schedule. If we have $t - z_k < 1$, the quantities $q_{k,t-z_k}$ do not denote decision variables, but scheduled receipts of product k from the implementation of previous production plans. Inequality (3) states the production quantities and setups must meet the capacity constraints for all the resources and inequality (4) ensures that a machine is set up for product k in period t if the product is produced during this period. Production quantities (5) and inventory levels (6) cannot be negative, the setup variable (7) is binary and there are neither initial nor ending inventories (8).

The MLCLSP is a mixed-integer linear program and can in principle be solved using a commercial solver. However, this optimization problem is NP-hard. Even the feasibility problem ("Does a solution exist?") is already NP-complete for the single-level problem if there are setup times. It is therefore not surprising that the use of commercial standard software for mixed-integer programming such as CPLEX is not very promising if the problem instances get large because the computation times explode. For this reason, we now describe a heuristic procedure to solve the MLCLSP which is based on a modification of the core equations (1) to (4) of the original model formulation.

The first step is to modify the MLCLSP by adding a virtual period 0 during which each resource has infinite capacity. This modified version of the MLCLSP is denoted as $\overline{\text{MLCLSP}}$. Note that while a given instance of the MLCLSP may or may not have a feasible solution, there is always a formally feasible solution to the problem $\overline{\text{MLCLSP}}$ because formally all the demand can be "produced" during period 0. Obviously, a production plan which actually contains positive production quantities planned for period 0 cannot be implemented and is therefore not desirable. For this reason, "production" during period 0 has to be penalized heavily to make sure that the optimization algorithm avoid production during period 0 if possible. Because of the multi-level structure of the production process, the penalty must take the preceding production levels into account. We used the penalty p_k for production of product k during period 0 which is defined recursively over the immediate predecessors $i \in \mathcal{V}_k$ of product k as follows:

$$p_k^0 := h_k + \max_i \{s_i\} + \sum_{i \in \mathcal{V}_k} p_i^0 \tag{9}$$

If we denote the production quantity of product k during the virtual period 0 as p_k^0, the model $\overline{\text{MLCLSP}}$ can be stated as follows:

Model $\overline{\textbf{MLCLSP}}$

$$\text{Minimize } \overline{Z} = \sum_{k=1}^{K} \sum_{t=1}^{T} (s_k \cdot \gamma_{kt} + h_k \cdot y_{kt}) + \sum_{k=1}^{K} p_k^0 \cdot q_k^0 \tag{10}$$

subject to

$$y_{k0} + q_{k,1-z_k} - \sum_{i \in \mathcal{N}_k} a_{ki} \cdot q_{i1} - y_{k1} + q_k^0 = d_{k1} \qquad \forall\, k, t = 1 \tag{11}$$

$$y_{k,t-1} + q_{k,t-z_k} - \sum_{i \in \mathcal{N}_k} a_{ki} \cdot q_{it} - y_{kt} = d_{kt} \qquad \forall\, k, t = 2, \ldots, T \tag{12}$$

and contraints (3)-(4) from the original MLCLSP formulation. Please note that if the penalty costs p_k for production during the virtual period 0 are sufficiently high such that in an optimal solution to problem $\overline{\text{MLCLSP}}$ no production quantities are planned for period 0, then this optimal solution to problem $\overline{\text{MLCLSP}}$ is also an optimal solution to the original problem MLCLSP.

Since our algorithm to solve the MLCLSP decides about setup patterns and production quantities separately, we now consider a variant of problem $\overline{\text{MLCLSP}}$ which assume that the setup pattern is given, i.e., that for all the binary setup variables γ_{kt} specific values $\overline{\gamma}_{kt}$ have "somehow" been determined. In this case, setup costs are no longer relevant and the problem reduces to minimize holding and penalty costs. This leads to the following model:

Model $\overline{\textbf{MLCLSP}}_{Fix}$:

$$\text{Minimize } \overline{Z}_{Fix} = \sum_{k=1}^{K} \sum_{t=1}^{T} h_k y_{kt} + \sum_{k=1}^{K} p_k^0 q_k^0 \tag{13}$$

subject to

$$y_{k0} + q_{k,1-z_k} - \sum_{i \in \mathcal{N}_k} a_{ki} \cdot q_{i1} - y_{k1} + q_k^0 = d_{k1} \qquad \forall\, k, t = 1 \tag{14}$$

$$y_{k,t-1} + q_{k,t-z_k} - \sum_{i \in \mathcal{N}_k} a_{ki} \cdot q_{it} - y_{kt} = d_{kt} \qquad \forall\, k, t = 2, \ldots, T \tag{15}$$

$$-\Big(\sum_{k \in \mathcal{K}_j} tp_k \cdot q_{kt}\Big) \geq -\Big(b_{jt} - \sum_{k \in \mathcal{K}_j} ts_k \cdot \overline{\gamma_{kt}}\Big) \qquad \forall\, j, t \tag{16}$$

$$-\, q_{kt} \geq -B \cdot \overline{\gamma}_{kt} \qquad \forall\, k, t \tag{17}$$

The problem $\overline{\text{MLCLSP}}_{Fix}$ is a linear program without any binary or integer decision variables since the given setup pattern only affects the right-hand side of constraints (16) and (17). Any setup pattern is a formally feasible solution if the capacity of all the machines is sufficient to allow for the planned setups in this pattern. Formally, the $\overline{\text{MLCLSP}}_{Fix}$ can also be stated in matrix notation:

$$\text{Minimize } Z(\underline{x}) = \underline{c}^T \underline{x} \tag{18}$$

subject to

$$\underline{D}_1 \underline{x} = \underline{g}_1 \tag{19}$$

$$\underline{D}_2 \underline{x} \geq \underline{g}_2 \tag{20}$$

$$\underline{x} \geq 0 \tag{21}$$

if we appropriately define the vector of objective function coefficients $\underline{c}$ from (13), the vectors $\underline{g}_1$ and $\underline{g}_2$ from the right-hand side of constraints (14) - (17) and the matrices $\underline{D}_1$ and $\underline{D}_2$ from the left-hand side of constraints (14) - (17). The vector $\underline{x}$ contains the (non-negative) production quantities and inventory levels. This linear program has a dual linear problem in decision variables $\underline{w}_1$ and $\underline{w}_2$ which can be stated as follows:

Modell $\overline{\textbf{MLCLSP}}^D_{Fix}$

$$\text{Maximize } ZD(\underline{w}) = \underline{g}_1^T \underline{w}_1 + \underline{g}_2^T \underline{w}_2 \tag{22}$$

subject to

$$\underline{D}_1^T \underline{w}_1 + \underline{D}_2^T \underline{w}_2 \leq \underline{c} \tag{23}$$

$$\underline{w}_1 \in \Re \tag{24}$$

$$\underline{w}_2 \geq 0 \tag{25}$$

A basic result from the theory of linear optimization is that the optimal objective function value $Z(\underline{x}^*)$ of the primal problem and the optimal value $ZD(\underline{w}_1^*, \underline{w}_2^*)$ of the corresponding dual problem are identical. In the optimal solution, the decision variables of the primal problem correspond to the shadow prices of the dual and vice versa. For this reason, we can either solve problem $\overline{\text{MLCLSP}}_{Fix}$ or $\overline{\text{MLCLSP}}^D_{Fix}$ to determine the optimal production quantities for a given setup pattern. The reason to work with the dual problem $\overline{\text{MLCLSP}}^D_{Fix}$ instead of the primal problem $\overline{\text{MLCLSP}}_{Fix}$ is that any change in the setup pattern does not affect the constraints of the dual problem. Therefore, any feasible solution of the dual problem remains feasible if we change the setup pattern. A change in the setup pattern only affects the coefficients of the objective function of the dual problem $\overline{\text{MLCLSP}}^D_{Fix}$. We therefore hope to be able to evaluate a new given setup pattern by doing a limited re-optimization of the dual problem $\overline{\text{MLCLSP}}^D_{Fix}$ which should take only a few iterations of the Simplex method.

3 A Solution Procedure Based on Dual Re-Optimization

The algorithm to solve the MLCLSP consists of an outer loop in which the setup patterns are modified (see Algorithm 1) and an inner loop in which a new current solution within the neighborhood of the current solution is selected (see Algorithm 2).

Algorithm 1 Outline of the tabu search algorithm

```
generate and evaluate initial solution with γ_kt := 1, ∀k, t
set intial solution as best solution so far
repeat
   generate randomly N candidate solutions that are not tabu
   select best candidate solution as the new current solution
   if new current solution is better than best solution so far then
      update best solution so far
   end if
   update tabu list
until maximum number of iterations is reached
```

Algorithm 2 Identification of the best candidate solution

```
modify objective function of Problem MLCLSP̄^D_Fix for the first candidate
determine the optimal solution of Problem MLCLSP̄^D_Fix for the first candidate
set this solution as the best candidate solution so far
for all other candidate solutions do
   modify objective function of Problem MLCLSP̄^D_Fix
   if the solution is better than the best candidate solution then
      update the best candidate solution
   end if
end for
```

In order to determine the production quantities and inventory levels for a given setup pattern, the linear program $\overline{\text{MLCLSP}}^D_{Fix}$ can be solved. In order to evaluate a neighborhood solution which differs with respect to its setup pattern in just one randomly selected and inverted setup variable γ_{kt}, it is not necessary to solve problem $\overline{\text{MLCLSP}}^D_{Fix}$ from scratch as the change of γ_{kt} affects only the objective function. It is therefore possible to use a dual re-optimization using the current basis of the linear program. This reduces the computational burden. In addition, it may be unnecessary to solve problem $\overline{\text{MLCLSP}}^D_{Fix}$ for a candidate solution to optimality. If during an iteration of the Simplex method the objective function of problem $\overline{\text{MLCLSP}}^D_{Fix}$ plus the setup costs related to the underlying setup pattern already exceeds the optimal value of a formerly treated best candidate so far, then the currently considered candidate cannot possibly be better and its problem $\overline{\text{MLCLSP}}^D_{Fix}$ does not have to be solved to optimality to reject this candidate. Note that the procedure to generate neighborhood solutions by inverting randomly selected setup variables is just a very simple first approach to the problem.

4 Preliminary Numerical Results

For a first evaluation of the proposed algorithm we compared its results to those obtained by other algorithms for a set of 75 problems introduced in [5] for the problem structure depicted in Figure 1. In these problems, three demand time series of end product demand with coefficients of variation of 0.1, 0.4 and 0.7 were combined with five setup cost profiles and five capacity/setup time profiles. The resulting problem instances were solved via a decomposition approach (DA) introduced in [4, 12], a simple simulated annealing algorithm (SAS), a more elaborate simulated annealing algorithm (SAE), a genetic algorithm (GA), a different tabu search algorithm (TS) and an algorithm based on evolution strategies

	ADLB [%]	MDLB [%]	ADUB [%]
DA	22.63	51.91	7.77
SAS	16.72	47.99	2.58
SAS-DA	16.99	48.76	2.77
SAE	15.14	46.67	1.20
SAE-DA	15.21	40.27	1.30
TS	19.20	51.43	4.73
TS-DA	16.99	45.83	2.81
GA	27.57	68.98	11.94
GA-DA	21.94	61.90	7.08
ES	18.75	54.67	4.20
ES-DA	17.65	60.02	3.31
TD	19.47	58.63	4.85
DROpt	19.26	49.91	4.71

Table 2: Numerical results

(ES), all described in [4]. In the variants SAE-DA, SAA-DA, TS-DA, GA-DA, and ES-DA, these local search procedures started from the solution obtained by the decomposition approach. In addition, the 75 problem instances were solved via a Lagrangean-based procedure (TD) developed by Derstroff and Tempelmeier, see [1, 10, 11] and the dual reoptimization (DROpt) presented in this paper. In [5], it had not been possible to compute optimal solutions through a MIP-solver even though the tight shortest-route based formulation of the MLCLSP [2, 4, 12] was used. It is, however, possible to compute lower bounds (LB) using the LP-relaxation of the shortest-route based formulation of the MLCLSP.

The numerical results are summarized in Table 2. The average and the maximum deviation from the lower bound (in percent of the lower bound) over 75 problems are denoted as ADLB and MDLB respectively. We did also determine the best feasible solution UB* (lowest known upper bound) over all 13 heuristically obtained solutions for each problem. The average deviation (UB-UB*)/UB* from the best known solution U* denoted as ADUB is a lower bound on the deviation from the optimum. More details about all the procedures but DROpt are reported in [4, 5]. We do not report computation times as computers with very different capabilities have been used to solve the problems and the new algorithm DR is in a very early stage of its development. The preliminary results indicate that the underlying approach of algorithm DROpt is promising. The algorithms DA and TD are highly problem-specific. In the local search algorithms, the procedure to solve the linear program for a given setup pattern is only a heuristic which is also highly problem-specific. In addition, some of the “old” local search procedures are rather time-consuming. The solution approach of the algorithm DROpt, however, is fairly general and can be be adapted to similar problem settings without much effort.

5 Conclusion

We have presented a new algorithm to solve the MLCLSP which combines a local search procedure with dual re-optimization via a standard solver. The algorithm is easy to describe and implement and it is flexible enough to consider aspects such as maximum inventory levels or parallel machines which cannot be easily incorporated into very problem-specific heuristic procedures. Future research will be directed at the improvement of the currently very simple optimization of the setup pattern, at more difficult variants of the MLCLSP and at extensive numerical studies.

References

[1] M.C. Derstroff. *Mehrstufige Losgrößenplanung mit Kapazitätsbeschränkungen*. Physica, Heidelberg, 1995.

[2] Gary D. Eppen and R. Kipp Martin. Solving multi-item capacitated lot-sizing problems using variable redefinition. *Operations Research*, 35(6):832–848, 1987.

[3] Bernhard Fleischmann and Herbert Meyr. The general lotsizing and scheduling problem. *Operations Research Spektrum*, 19:11–21, 1997.

[4] Stefan Helber. *Kapazitätsorientierte Losgrößenplanung in PPS-Systemen*. M&P Verlag für Wissenschaft und Forschung, Stuttgart, 1994.

[5] Stefan Helber. Lot sizing in capacitated production planning and control systems. *Operations Research Spectrum*, 17:5–18, 1995.

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

[7] Herbert Meyr. *Simultane Losgrößen- und Reihenfolgeplanung für kontinuierliche Produktionslinen*. Gabler, Wiesbaden, 1999.

[8] Herbert Meyr. Simultaneous lotsizing and scheduling by combining local search with dual reoptimization. *European Journal of Operational Research*, 120:311–326, 2000.

[9] Horst Tempelmeier. *Material-Logistik. Modelle und Algorithmen für die Produktionsplanung und -steuerung und das Supply Chain Management*. Springer, Berlin, 5 edition, 2003.

[10] Horst Tempelmeier and Matthias Derstroff. Mehrstufige Mehrprodukt-Losgrößenplanung bei beschränkten Ressourcen und genereller Erzeugnisstruktur. *Operations Research Spectrum*, 15:63–73, 1993.

[11] Horst Tempelmeier and Matthias Derstroff. A lagrangean-based heuristic for dynamic multi-level multi-item constrained lotsizing with setup times. *Management Science*, 42:738–757, 1996.

[12] Horst Tempelmeier and Stefan Helber. A heuristic for dynamic multi-item multi-level capacitated lotsizing for general product structures. *European Journal of Operational Research*, 75:296–311, 1994.