A Stochastic Lot-sizing and Scheduling Model

Sven Grothklags[1], Ulf Lorenz[1], and Jan Wesemann[2]

[1] Institut für Informatik, Paderborn University
[2] Wirtschaftsinformatik, insb. CIM, Paderborn University

Companies are trying to improve their ability to deliver and to reduce their production costs simultaneously. Nevertheless ongoing orders of customers are causing difficulties to deliver because of bottlenecks in the production. Furthermore constantly incoming customer orders effect that every actual optimal production plan with its already realized and planned guidelines has induced states that are not anymore ideal for further production. In addition, current production planning tools neglect the continuation of the production beyond the planning horizon. Our purpose is to develop a new tool for production planning. As a first step, we propose a production planning model which includes scheduling, lot-sizing, dynamic demand, sequence dependent setup times, backlogging, stockouts, parallel machines, and, last but not least, a careful injection of stochastic data.

1 Introduction

In the academic world, as well as in Industry, there is a wide unity that companies with a combination of high customer-value products and good logistics are especially successful in competition. Good logistics are marked by low production- and inventory costs as well as by a high ability to deliver timely.

Although the benefi t of good logistics is indisputable, there are quite different ideas, how to achieve the goal. One approach begins with the formulation of a mathematical optimization model, which monolithically describes the planning system. The model allows to determine and to compare different admissible solutions. Moreover, various solution procedures and algorithms can be evaluated in a systematical way. A weakness of the currently published models is that input data are typically assumed to be deterministic. Approaches which combine dynamic capacitate planing with considering stochastic influences are still missing [13]. A different, however, very important approach with its emphasis more on controlling than on planning, mainly used for stock production can e.g. found in [8].

We present a new model, based on a direction as proposed e.g. in [13, 6, 2]. Injecting a certain amount of stochasticity into the data of such model, leads us to the fi elds of Decision Making Under Uncertainty [11], On line Algorithms [5], Stochastic Optimization [4] etc. Recently, we saw that stochastic information can indeed be exploited [1, 3, 7, 9, 12]. All these stochastic approaches fall into a category which Papadimitriou called 'Games against Nature' [10]. Interestingly, the typical complexity class of these games is PSPACE, the same complexity class to that many two-person zero-sum games belong.

In the next section, we elaborate such a model for production planing, and thereafter, we give a short outlook.

2 Production Planning as a Game against Nature

2.1 The Deterministic small-bucket multi-level capacitated lot-sizing and scheduling with parallel machines (D-SMLCLS-PM)

We are interested in a combined problem of lot sizing and scheduling with the following properties: Several items have to be produced to meet some known dynamic demand. A general gozinto-network provides the 'is-piece-of'-relations between the items. Backlogging, stockouts, and positive lead times are considered.

The production of an item requires the exclusive access to a machine, and items compete for these machines. A specifi c item may be produced alternatively on different machines and can have different production speeds on single machines. Even the production speed on the same machine can vary over time.

The production of an item on a machine can only take place if the machine is in a proper state. Changing the setup state of a machine from one state to another causes setup costs and setup times, which both may depend on the old and new state. Setup states are kept until another setup changes occur.

The planning horizon is divided into a fi nite number of time periods. Items which are produced in a period to meet some future demand must be stored in an inventory and cause item-specifi c holding costs.

One restriction which we invented to the model is that setup changes are only allowed to start at the beginning of a period. As a consequence, every machine can produces at most one item type per period.

The following variables are used to encode a solution:

Symbol	Definition
$q_{i,t}$	Production quantity of item P_i by task i in period t
$s_{p,t}$	Inventory for item p at the end of period t
$b_{p,t}$	Backlog of item p at the end of period t
$o_{p,t}$	Shortfall quantity of item p in period t (no backlog)
$x_{i,j,t}$	Binary variable which indicates whether a setup change from task i to j occurs in (at the begin of) period t ($x_{i,j,t} = 1$) or not ($x_{i,j,t} = 0$)
$y_{i,t}$	Binary variable which indicates whether machine M_i is set up for task i at the end of (during) period t ($y_{i,t} = 1$) or not ($y_{i_t} = 0$)

To describe the optimization model we also need the following input parameters:

Symbol	Definition
P	Set of items
M	Set of machines
I	Set of tasks
T	Number of periods $(1 \dots T)$
M_i	Machine needed by task i
P_i	Item produced by task i
I_i	$= \{j \in I : M_j = M_i, j \neq i\}$; Set of tasks using the same machine as task i
S_p	Nonnegative holding cost for having one unit of item p one period in inventory
B_p	Nonnegative cost for backlogging one unit of item p for one period
O_p	Stockout cost for not fulfilling the (external) demand of one unit of item p
$X_{i,j}$	Nonnegative setup cost for switching from task i to j (on machine $M_i = M_j$)
Q_i	Production cost for one unit of item P_i with task i
$D_{p,t}$	External demand for item p in period t
$A_{p,q}$	Gozinto-factor. It is zero if item q is not an immediate successor of item p. Otherwise: the quantity of item p that is directly needed to produce one item q
$\Delta_{p,i}$	(Integral) number of periods for transporting item p to machine M_i (for task i)
$C_{i,t}$	(Maximum) production quantity of item P_i with task i during period t
$T_{i,j}$	(Fractional) number of periods for switching from task i to j
$R_{i,j}$	$= \lfloor T_{i,j} \rfloor$; Number of periods exclusively used for switching from task i to j
$R^f_{i,j}$	$= T_{i,j} - R_{i,j}$; Fraction of the the $(R_{i,j}+1)$th period after switching from task i to j used for switching

Now, the following MIP defi nes our lot sizing and scheduling problem.

$$\text{minimzie} \sum_{t=1}^{T} \left(\sum_{p \in P} (S_p s_{p,t} + B_p b_{p,t} + O_p o_{p,t}) + \sum_{\substack{i \in I \\ j \in I_i}} X_{i,j} x_{i,j,t} + \sum_{i \in I} Q_i q_{i,t} \right) \tag{1}$$

subject to

$$s_{p,t-1} + \sum_{i:P_i=p} q_{i,t} - D_{p,t} - \sum_{\substack{q \in P \\ j:P_j=q}} A_{p,q} q_{j,t+\Delta_{p,j}} + b_{p,t} + o_{p,t} - b_{p,t-1} = s_{p,t} \qquad \forall p \in P,\ t = 1 \dots T \tag{2}$$

$$b_{p,t} + o_{p,t} - b_{p,t-1} \leq D_{p,t} \qquad \forall p \in P,\ t = 1 \dots T \tag{3}$$

$$\sum_{i:M_i=m} y_{i,t} = 1 \qquad \forall m \in M,\ t = 1 \dots T \tag{4}$$

$$y_{i,t-1} + y_{j,t} - 1 \leq x_{i,j,t} \qquad \forall i \in I, j \in I_i,\ t = 1 \dots T \tag{5}$$

$$C_{i,t} \left(y_{i,t} - \sum_{\substack{j \in I_i \\ t-R_{j,i} < t' \leq t}} x_{j,i,t'} - \sum_{j \in I_i} R^f_{j,i} x_{j,i,t-R_{j,i}} \right) \geq q_{i,t} \qquad \forall i \in I,\ t = 1 \dots T \tag{6}$$

$$q_{i,t} \geq 0;\ s_{p,t},\ b_{p,t},\ o_{p,t} \geq 0 \qquad \forall i \in I, \forall p \in P\ \ t = 1 \dots T \tag{7}$$

$$x_{i,j,t} \in \{0,1\};\ y_{i,t} \in \{0,1\} \qquad \forall i \in I, j \in I_i,\ t = 1 \dots T \tag{8}$$

Objective (1) minimizes the sum of the total holding, backlog, stockout, setup, and production costs.

Equations (2) are the inventory balances. The inventory of item p at the end of period t can be expressed by the inventory at the end of period $t-1$ plus what was produced during period t minus external and internal demand. Furthermore, we must add the amount of items that are backlogged and canceled in period t (these items reduce the external demand in the current period) and subtract the number of items backlogged in period $t-1$ (these items must still be produced - now or in some future period). Note that for the internal demand, positive lead times $\Delta_{p,j}$ are taken into account by taking item p out of the inventory $\Delta_{p.j}$ periods before starting the demanding task j. No inventory costs for the internal demand of item p occur during these $\Delta_{p,j}$ periods but these costs can be added to the production costs of task j, if desired.

Inequalities (3) ensure that the number of items (additionally) backlogged and canceled does not exceed the external demand during each period. Therefore, the internal demand can always be fulfilled and production is guaranteed.

Equations (4) together with (8) ensure that every machine has exactly one setup state at the end of every period.

Inequalities (5) together with (8) identify setup changes. If a machine was in state i at the end of period $t-1$ ($y_{i,t-1} = 1$) and the state at the end of period t is j ($y_{j,t} = 1$) a setup change from i to j occurred during period t and inequality (5) forces $x_{i,j,t}$ to one. For all other setup states $x_{i,j,t}$ can become zero and positive setup costs $X_{i,j}$ will enforce this for optimal solutions. Note that therefore conditions (8) can be replaced by $x_{i,j,t} \geq 0$ without changing the set of optimal solutions.

Production capacity constraints are expressed by inequalities (6). First of all, a task i can only produce items in period t if machine M_i is in the proper setup state during this period ($y_{i,t} = 1$). If all occurring $x_{i,j,t}$ variables are zero we can then produce up to $C_{i,t}$ items in period t. As we want to take setup times into account in our model, we must also make sure that the setup change from some task j to i has been finished before (or during) period t. The first sum effectively reduces the available production capacity to zero for periods which are fully reserved for setup changes. The second sum cuts of the necessary fraction of the production capacity for periods during which a setup change finishes. Note that because of (7) the left hand side of an inequality of type (6) must not become negative. Thus all $x_{i,j,t}$ variables must be zero if $y_{i,t}$ is zero and at most one $x_{i,j,t}$ variable can be one if $y_{i,t}$ is one. This ensures that a machine remains in its setup state i during the whole setup change leading to state i and that no two setup changes can overlap.

The constraints (7) are the non-negativity conditions for production quantity, inventory, backlogging, and stockout variables. Constraints (8) define our binary-valued setup change and setup state variables.

With a bit more effort, the model can be extended in such a way that integral lot sizes are modeled as well.

2.2 The Stochastic small-bucket multi-level capacitated lot-sizing and scheduling with parallel machines (S-SMLCLS-PM)

On the basis of the previous section, we can now move from the D-SMLCLS-PM to a stochastic version, the S-SMLCLS-PM. In contrast to the deterministic version, we now assume that customer demands are not described with a natural number, but with the help of a finite probability distribution, which reflects that customer demands are not exactly known at planning time. Moreover, the production speed of the machines is described with the help of a probability distribution, as well.

We formalize the stochastic planning problem as a game. Before we go into details, let us inspect the notion of time. Let T_0 be the start time, T_e the end time of our game, T_i any time point between T_0 and T_e. Similar as in the deterministic case, the time is divided into intervals, into small time periods. Let the timespan between T_i and T_{i+1} be δ. We assume that everything that takes place between time T_i and $T_i+\delta$ happens simultaneously, at time point $T_i + \delta$. Let τ_{det} be a time period, such that we can assume demands with deadlines between T_i and $T_i + \tau_{det}$ to be deterministic. τ_{det} is a multiple of δ. Summarizing, we assume two different time blocks: deterministic knowledge about future external demands shortly behind the here-and-now time-line, as well as the sizes of past lots. Moreover, we assume to know finite distributions about demand-data beyond $T_i + \tau_{det}$, and finite distributions about future production capacities per period, i.e. the $C_{i,t}$.

The game has two players who move in an alternating way. One player is a company that tries to maximize its profits. The other player is Nature, who realizes probability distributions, rolls a die, respectively.

The actions of Nature can be described as follows. At time T_i, Nature determines the size of those lots which have been finished between T_i and T_{i-1}. Moreover, it determines those demands, whose deadlines are between T_i and $T_i + \tau_{det}$.

The company has to assign tasks to the machines ($=$ provide a plan) at least from time T_i to T_{i+1} at each time point T_i. The aim is to maximize the profit over the full period T_0 to T_e. Thus, instead to provide one plan, the company provides a sequence of plans $P_0, \ldots, P_e$ from time T_0 to time T_e. Obviously, the optimization task is a multi-stage decision process. Gains and costs of a realized run are gathered concerning Equation 1. Here, we see the relation between the stochastic and the deterministic optimization problems. From a posteriori point of view, the originally fuzzy input data have been determined and each run has to follow the production rules of deterministic problem description. Moreover, the costs and profits are determined with the help of the deterministic model. Note, however, that because of imperfect information at decision time, an optimal decision at a time point T_i is not necessarily identical with the optimal decision under the assumption that data realize as they do. An optimal decision at a time point T_i can only be optimal in expectation.

As soon as Nature has made a move, the company has to re-act. Ideally, the company instantly reacts.

3 Conclusion and Future Work

We have introduced a new production model which is mainly geared to the deterministic lotsizing and scheduling models for deterministic planning. Stochastic influences are additionally injected and as a result, we described the optimization problem in form of a Game against Nature.

The S-SMLCLS-PM-model allows to compare algorithms experimentally in a clearly defined optimization world. E.g., we can keep a certain, especially interesting problem instance fixed, and then make a lot of simulation runs. Because the input data are probability distributions with nice properties, we can conclude to the quality of an algorithm via statistics. Another way to evaluate an algorithm is to arrange competitions as we know them from game playing. We are going to implement a simulation world for the S-SMLCLS-PM-model, which allows to compare different algorithms with the help of statistics and with the help of competitions. We aim at realistic test instances like a complete motor of a car.

References

1. R. Bent and P. van Hentenryck. Scenario based planning for partially dynamic vehicle routing problems with stochastic customers. *Operations Research*, 52(6), 2004.
2. A. Drexl and K. Haase. Proportional lotzising and scheduling. *International Journal of Production Economics*, 40:73–87, 1995.
3. J. Ehrhoff, S. Grothklags, and U. Lorenz. Parallelism for Perturbation Management and Robust Plans. *Proceedings of the 11th International Euro-Par Conference, (J.C. Cunha, P.D. Medeiros eds), Springer LNCS 3648*, pages 1265–1274, 2005.
4. S. Engell, A. Märkert, G. Sand, and R. Schultz. Production planning in a multiproduct batch plant under uncertainty. *Preprint 495-2001, FB Mathematik, Gerhard-Mercator-Universität Duisburg*, 2001.
5. A. Fiat and G. Woeginger. Online Algorithms: The State of the Art. *LNCS*, 1998.
6. A. Kimms and A. Drexl. Proportional lot sizing and scheduling: Some extensions. *Networks*, 32:85–101, 1998.
7. A. Kleywegt and A. Shapiro. The sample average approximation method for stochastic discrete optimization. *SIAM Journal on Optimization*, 15(1-2), 1999.
8. H. Lödding. Verfahren der Fertigungssteuerung. *Springer, Berlin*, 2004.
9. R. Moehring, A. Schulz, and M. Uetz. Approximation in stochastic scheduling: the power of lp-based priority policies. *Journal of ACM*, 46(6):924–942, 1999.
10. C. H. Papadimitriou. Games against nature. *Journal of Computer and System Science*, 31:288–301, 1985.
11. A. Scholl. Robuste Planung und Optimierung. *Physiker Verlag*, 2001.
12. D. Shmoys and C. Swamy. Stochastic Optimization is (almost) as easy as Deterministic Computation. *Proceedings of Foundations in Computer Science (FOCS-2004)*, pages 228–237, 2004.
13. H. Tempelmeier. Material-Logistik, Modelle und Algorithmen für die Produktionsplanung und -steuerung in Advanced Planning-Systemen. *Springer, 6. Aufl., Berlin*, 2006.