A Stochastic Lot-Sizing and Scheduling Model

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

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

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 benefit 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 fields 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 specific 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 finite 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-specific 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 $(1 \ldots T)$
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