Sequencing and lot-size optimisation of a production-and-inventory-system with multiple items using simulation and parallel genetic algorithm

Michael Kämpf[1] and Peter Köchel[1]

University of Technology, Department of Computer Science, Chemnitz D-09107, Germany

Abstract Our paper is dealing with the Capacitated Stochastic Lot-Sizing Problem. In addition to the usual model assumptions as stochastic demand and manufacturing times, cost for setup, we also consider cost for waiting and lost demand. The goal is to find release and sequencing decisions with minimal expected cost per time unit. To solve the problem we use simulation optimisation, i. e., we combine a simulator with a Parallel Genetic Algorithm. Some numerical examples show the applicability of the proposed approach.

1 Introduction

One of the most important problems for manufacturing firms is production lot-sizing and sequencing. At present different problem formulations exist: ELSP, Economic Lot Scheduling Problem; SELSP, Stochastic Economic Lot Scheduling Problem; CLSP, Capacitated Lot-sizing and Scheduling Problem; CSLSP, Capacitated Stochastic Lot-sizing and Scheduling Problem. For a literature overview for all problem formulations see Kuik et al. (1994).

In our paper we investigate the simulation optimisation approach to a generalised formulation of the CSLSP (Kämpf and Köchel (2004)). Some main characteristics are: N items, a single manufacturing unit, item dependend setup times and cost, arbitrary demand processes, finite space in the storage, backordering and rejection of demand.

2 Model and problem formulation

The model we will investigate consists of various parts (see Fig. 1). We assume continuous time and infinite horizon.

The ***demand process*** is modeled as a compound renewal process with λ_n as intensity of arriving clients and B_n as random variable for the demand size of a client, $\forall n$. If there is no inventory on hand arriving demand can be backordered in a queue with capacity $0 \leq b_n \leq \infty$, $\forall n$. Clients who meet a full backorder queue will be rejected.

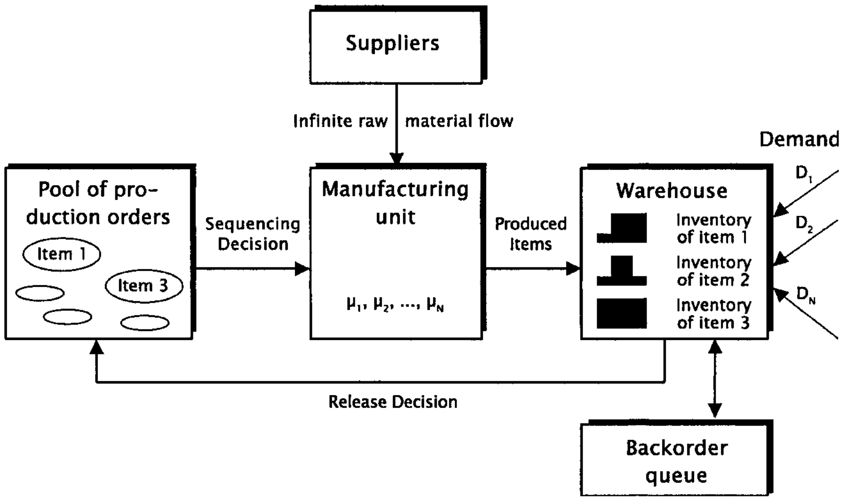

Figure1. Scheme of the production-and-inventory system

Produced items are used to satisfy backordered demand. If there is no waiting client the produced items will be stored in the ***warehouse.*** Each item occupies a defined size p_n in the warehouse. The warehouse itselves has capacity P with $0 \leq P \leq \infty$. In dependence of the release policy the warehouse sends a production order to the production pool.

The ***manufacturing unit*** works without failures. Item n is produced with rate μ_n, where it needs a random time to manufacture one unit. The only reasons why the production process must stop are no place for the produced item in the warehouse or emptiness of the pool of production orders.

Next we introduce various policies for the release of manufacturing orders and their sequencing in the production process. The corresponding decisions are called *release decision* and *sequencing decision.*

The production ***sequencing decision*** chooses from the pool of manufacturing orders the next one for manufacturing. A concrete decision is defined by a sequencing rule or sequencing policy. Some examples are the following ones:

First come first serve (FCFS): The production order first inserted in the pool will be produced next. It's denoted by P_S^{FCFS}.

Random: The next production order will be chosen randomly (P_S^{Random}).

Cyclic: There exist a precalculated cyclic production sequence $P_S^{Cyclic} = \{c_1, c_2, \ldots, c_N\}$. If the pool containts no production order for the currently choosen item the next position in the cycle will be considered and so on. In our case the order is predefined as $P_S^{Cyclic} = \{1, 2, \ldots, N\}$ without any claim of optimality.