A local search algorithm for the optimization of the stochastic economic lot scheduling problem

Citation for published version (APA):

Document status and date:
Published: 01/01/2002

Document Version:
Publisher’s PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:
• A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
• The final author version and the galley proof are versions of the publication after peer review.
• The final published version features the final layout of the paper including the volume, issue and page numbers.

Link to publication

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

• Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
• You may not further distribute the material or use it for any profit-making activity or commercial gain
• You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the “Taverne” license above, please follow below link for the End User Agreement:
www.tue.nl/taverne

Take down policy
If you believe that this document breaches copyright please contact us at:
openaccess@tue.nl
providing details and we will investigate your claim.

Download date: 19. Mar. 2020
A local search algorithm for the optimization of the stochastic economic lot scheduling problem
Michael Wagner and Sanne R. Smits
WP 76

<table>
<thead>
<tr>
<th>BETA publicatie</th>
<th>WP 76 (working paper)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ISBN</td>
<td>90-386-1827-1</td>
</tr>
<tr>
<td>ISSN</td>
<td>1386-9213</td>
</tr>
<tr>
<td>NUR</td>
<td>804</td>
</tr>
<tr>
<td>Eindhoven</td>
<td>May 2002</td>
</tr>
<tr>
<td>Keywords</td>
<td>Stochastic economic lot scheduling problem / Local search</td>
</tr>
<tr>
<td>BETA-Research Programme</td>
<td>Chain Management</td>
</tr>
<tr>
<td>Te publiceren in:</td>
<td>.</td>
</tr>
</tbody>
</table>
A Local Search algorithm for the optimization of the stochastic economic lot scheduling problem

Michael Wagner and Sanne R. Smits

April 12, 2002

Michael Wagner, University of Augsburg, Department of Production and Logistics, Universitaetsstr. 16, D-86135 Augsburg, Germany
Michael.Wagner@wiwi.uni-augsburg.de

Sanne R. Smits, Faculty of Technology Management, Technische Universiteit Eindhoven, Paviljoen E4, P.O. Box 513, 5600 MB Eindhoven, The Netherlands
S.R.Smits@tm.tue.nl
Abstract

This paper describes a model for the stochastic economic lot scheduling problem (SELSP) and a Local Search heuristic to find close to optimal solutions to this model. The SELSP considers multiple products, which have to be scheduled on a single facility with limited capacity and significant setup times and costs. The demand is modeled as stationary compound renewal process. The objective is to find a schedule that minimizes the long-run average costs for setups and inventories while satisfying a given fill rate. We use a cyclic scheduling approach in which the individual cycle time of each product is a multiple of some basic period (fundamental cycle).

For the deterministic version of the SELSP efficient heuristics have been developed which guarantee the feasibility of the solution by adding an additional constraint to the problem. In our case this is not sufficient, because for the calculation of the average inventory levels and fill rates we need to develop a schedule with detailed timing of the lots. We present an efficient heuristic for this scheduling problem, which can also be used to check the feasibility of the solution. Thereby, the most time consuming step (the calculation of average inventory levels and fill rates) is only performed for a limited set of candidates.

The algorithm was tested on deterministic benchmark problems from literature and on a large set of stochastic instances. We report on the performance of the heuristic in both cases and try to identify the main factors influencing the objective.

Keywords: Stochastic economic lot scheduling problem, Local Search

1 Introduction

Recently stochastic models for production planning gain importance in research and also in practice. We will introduce a planning model which is applicable for 1-stage capacitated multi-product production systems. These systems can be found in the process industries, especially in consumer goods industries. Our model assumes stochastic but stationary demand, which is modeled as compound renewal process. The changeovers between different products require significant setup times and costs. Therefore the objective of the problem is to schedule the production such that the
sum of setup costs and inventory holding costs is minimized. The deterministic version of the problem is known as the Economic Lot Scheduling Problem (ELSP) and has attracted many researchers. The ELSP is known to be NP-hard (see Hsu (1983)) and therefore most solution approaches proposed in literature are specific heuristic algorithms.

The stochastic version of the ELSP (Stochastic ELSP, SELSP) has been addressed only recently. In addition to the deterministic ELSP it is necessary to include the problem of backorders or lost sales either by penalty costs or by a service level constraint. Furthermore, it is necessary to define how dynamically changing demand is handled in such a stationary model. In a recent review on the SELSP Sox et al. (1999) distinguish the independent stochastic control and the joint deterministic control. Both methods try to implement cyclic approaches in a stochastic environment. In the independent stochastic control approach single product inventory control policies (e.g. \((s,S)\) or \((s,Q)\)) are used to generate the quantity and release time of production orders. In this case the inventory control parameters are calculated using lead times, which can be derived from queuing models. The joint deterministic control approach constructs the schedule and the inventory control policy simultaneously based on deterministic demand. Afterwards safety stocks are integrated as minimum stock level to guarantee a certain service level. This approach does not integrate the safety stocks in the scheduling algorithm and therefore ignores the dependencies of the schedule and the level of safety stocks. In our model we will use the joint deterministic control approach, but integrate the safety stock in the objective function explicitly. This allows us to optimize on all relevant cost components. The calculation of safety stock levels requires the determination of a detailed schedule, because the production lead time depends on all production lots of a cycle.

In contrast to the classical ELSP model, which assumes continuous availability of the production quantity, we assume batch availability. This means, that the total production lot is not available until the end of the production run.

There exists a rich literature on the optimization of the deterministic ELSP (for recent review see Zipkin (1991)). Most of the solution approaches limit the number of potential solutions by introducing additional constraints which facilitate the search.
for feasible solutions. Bomberger (1966) introduced the Basic Period approach in which all cycles are integer multiples of some basic period. Furthermore, he postulated that the basic period must be long enough to accommodate the setup times and production lots of all items. As this additional constraint limits the solution space very heavily, Elmaghraby (1978) Doll and Whybark (1973) and Goyal (1975) relaxed or removed the constraint to admit solutions with lower costs. Most of these heuristic approaches lack a rigorous method for generating a feasible schedule, which is one of the hardest parts of the solution process. This drawback is circumvented by the algorithms of Geng and Vickson (1988) and Haessler (Haessler and Hogue (1976), Haessler (1979)). They provide mechanisms to check for feasibility and enable fast construction of schedules by restricting the item multipliers to powers of two. The heuristics successively determine the length of the basic period and the item multipliers until they converge. Among this class of algorithms the best solutions are reported for the Doll & Whybark procedure and the extended version of Haessler and Hogue (1976).

Only recently also evolutionary approaches are reported for the ELSP based on the basic period model. Tunasar and Rajgopal (1996) use the Local Expansion Search Heuristic (LES) to find the best combination of item multipliers and calculate the optimal basic period for it based on a formula proposed by Doll and Whybark (1973). The LES combines elements of genetic algorithms and local search to overcome drawbacks of each method. For the feasibility check they provide a simple heuristic, which tries to schedule the items in a one-pass procedure. This fast heuristic enables the evaluation of a large amount of different solution candidates in a short time. The optimization algorithm is tested on some benchmark problems (with and without power-of-two restriction) and is reported to be superior compared to previous approaches. Khouja et al. (1998) provide a genetic algorithm to solve the ELSP using the basic period formulation. They discuss various ways of coding the solution and different crossover operations, which are also tested on some benchmark problems. Especially they code the real valued basic period in binary format to be able to use genetic operations on the solution. Infeasible solutions are penalized by a multiple of the capacity overrun, which is increased during the solution process. The capacity
overrun can be calculated easily in their case, because their model is based on the original basic period approach (Bomberger (1966)), which requires all products to be produced in one basic period. This constraint prevents their algorithm from being better than the approach of Tunasar and Rajgopal (1996) and Haessler (1979).

For the case of the stochastic version of the ELSP the literature is rather limited and often based on restrictive assumptions. These assumptions include models with only two items or without setup time or costs. In the following we will review only papers, which are based on practical assumptions and applicable models. Zipkin (1986) and Karmarkar (1987) present models, which are of the independent stochastic control type introduced above. In their models the items are controlled by single product inventory control policies, which generate production orders that are processed on a first-come-first-served basis. Models for the calculation of inventory control parameters in this case are reviewed in a paper of Lambrecht and Vandaele (1996). The class of joint deterministic control models (or semi-dynamic models (Federgruen and Katalan (1996))) was introduced by Gallego (1990) and Gallego (1994). He uses backorder costs and proposes a recovery policy to keep up with the target schedule determined initially. The inventory control parameters have to be computed by Monte Carlo simulation because of the complex structure of the recovery policy. The same applies to the model of Bourland and Yano (1994), which is based on a pure rotation cycle (each product is produced only once in the cycle) and uses overtime in shortage situations. Thus, the cost function comprises inventory, setup and overtime costs, which should be minimized in the planning model. To our knowledge the only SELSP model which can be evaluated analytically was published by Federgruen and Katalan (1996) and Federgruen and Katalan (1998). Their model is based on Poisson-distributed demand, backorder costs and is controlled by a base-stock policy. The solution approach consists of three phases:

1. Calculate production frequencies for each item.

2. Determine the length of the production table.

3. Sequence the production lots in the table.
Federgruen and Katalan (1998) develop algorithms for each the phases listed above and proof the overall efficiency of the procedure. The approach is similar to the basic period model as the schedule also consists of multiples (frequencies) and a basic period (length of the table). Their model and ours differ in the way the inventory is controlled: They continue production until a given base-stock is reached, whereas our model generates orders from a fixed schedule. The SELSP formulation of Federgruen and Katalan (1998) can be analyzed by so-called polling models (see e.g. Westrate (1992)), which are queuing systems with multiple queues. In contrast to the fixed schedule models the polling models are characterized by a high variability in utilization.

The remainder of this paper is organized as follows. In Section 2 we introduce the production scheduling and inventory control policy used for the SELSP and give a brief introduction on how the control parameters can be calculated for a given schedule. Section 3 describes the Local Search algorithm used to optimize the schedule. Results of a numerical study are given in Section 4 and the paper finishes with a short conclusion and outlook on further research.

2 Model description

We consider a single capacitated production facility which produces \( N \) different items with independent stochastic demand. For changeovers from one item to another significant sequence-independent setup costs and times occur and the item specific production rates are linear and deterministic. The demand process is stationary and can be modeled by a compound renewal process with known parameters for the probability distributions of the inter-arrival rate \( A_i \) and the order size \( D_i \) of an arbitrary customer. Both components are mixed-Erlang distributed and the demands of independent items are uncorrelated. Since the production quantity is only available after the whole lot is finished (batch availability) and therefore, the lead time of a production order consists of the setup time, the production time of the complete lot and a potential waiting time. Unsatisfied demand is backordered and the customer service is measured by the fill rate. The fill rate states the expected
fraction of demand that can be delivered directly from stock.

The functions and operators used in the following are summarized in Table 1:

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(D(Y))</td>
<td>Demand during time (Y)</td>
</tr>
<tr>
<td>(E[Y])</td>
<td>Expectation of the random variable (Y)</td>
</tr>
<tr>
<td>(\sigma^2(Y))</td>
<td>Variance of the random variable (Y)</td>
</tr>
<tr>
<td>((a)^+)</td>
<td>Maximum of the real number (a) and zero</td>
</tr>
</tbody>
</table>

We suggest a two-level solution approach to the SELSP. First, a cyclic schedule \(\Omega\) is created which minimizes the long-run expected costs for setups and inventories (planning level). The schedule is repeated after the complete rotation cycle \(T\), which equals the maximum of all item specific cycle times \(t_i\). All cycle times are integer multiples \(k_i \in \{2^0, 2^1, 2^2, \ldots \}\) of a basic period \(T^{BP}\).

Second, a short-term control policy has to specify how the static cyclic schedule can be used in a dynamic environment (control level). In our model we generate production orders using a periodic order-up-to inventory policy. The review period \(R\) of the \((R,S)\)-policy is given by the individual cycle times \(t_i\) of each product. Each time the stock of an item is reviewed, a production order is issued that brings the inventory position back to the order-up-to level \(S\). The review periods are given by the schedule \(\Omega\) calculated before (release schedule). The production orders are processed on a first-come-first-served (FIFO) basis and cannot be discontinued. Thus, each order has to wait until all preceding orders are finished and the machine can be set up for the next one. If no order is waiting for processing, the facility remains idle until the next job is generated (production schedule).

We describe the schedule \(\Omega\) by a fixed sequence of \(K\) production intervals \(k\) which consist of the setup time \(s_{v(k)}\) for item \(v(k)\), the production time and possibly some idle time before the next item is setup. Each item can be scheduled more than once per cycle and therefore we count the number of setups for each item from the beginning of the cycle by index \(j\). The total length of the production interval \(k\) is denoted by the variable \(U_k\) and the production order size is \(Q_k\). The production time for interval \(k\) can be derived from \(Q_k/\rho_{v(k)}\), where \(\rho_{v(k)}\) is the production rate. In the following we will show how a given schedule can be analyzed and appropriate
Figure 1: The two-level solution approach

order up-to levels $S_{j,i}$ can be calculated for each setup $j$ for item $i$. We give only a brief overview on the analytical part of the model which is described in detail in Smits and Wagner (2002). The order-up-to levels have to guarantee a given fill rate $\beta_i$ and might vary depending on the position in the overall cycle, because the lead time $L_k$ for an order includes not only the setup and the production time, but also an interval dependent waiting time $W_k$. The total lead time is given by:

$$L_k = W_k + Q_k/\rho_{v(k)} + s_{v(k)}$$  \hspace{1cm} (1)

For the analysis of order-up-to levels we need to estimate the first two moments of
the lead time $L_k$. For this we model the production system as a D/G/1-queue and derive approximations for the waiting time and service time of the queuing system. The arrival process of the customers (production orders) is deterministic, because it follows the cyclic schedule, but the service time $B_k$ is stochastic because of the stochastic order size $Q_k$. The order size equals the item demand during the last review period $R_{v(k)}$ and will be denoted by $D(R_{v(k)})$.

$$B_k = D(R_{v(k)})/\rho_{v(k)} + s_{v(k)}$$

We fit a mixed-Erlang distribution on the service time based on the first two moments (see Tijms (1994)). This is an approximation, because subsequent order sizes $Q_k$ are not independent. If we know the service time distribution we can describe the waiting time of each order by Lindley’s equation:

$$W_{k+1} = (W_k + B_k - U_k)^+ \quad \forall k = 1, \ldots, K - 1, \text{and}$$

$$W_1 = (W_K + B_K - U_K)^+$$

Then we use a moment-iteration method first proposed by de Kok (1989) to approximate the first two moments of the waiting time $W_k$. This algorithm converges in only a few iterations and is reported to yield quite accurate results for the waiting time distribution (see Smits and Wagner (2002)). Using these results, the first two moments of the lead time of the $f(k)$th setup of item $v(k)$ are given by:

$$E[L_{f(k),v(k)}] = E[W_k] + E[B_k]$$

$$\sigma^2(L_{f(k),v(k)}) = \sigma^2(W_k) + \sigma^2(B_k)$$

The calculation of the order-up-to levels is based on the analysis of the expected shortage per review period. The following definition of the fill rate is used in most textbooks on inventory management (e.g. Silver et al. (1998)) and can be applied here as well.

$$\beta_i = 1 - \frac{E[(-I_{e(j,i)})^+] - E[(-I_{b(j,i)})^+]}{E[D(R_t)]}$$

We derive the expected shortage from the expected net inventory at the beginning of the cycle $I_{b(j,i)}$ and the net inventory at the end of the cycle $I_{e(j,i)}$. These values
are calculated by using the following expressions:

\[ I_{e(j,i)} = S_{j,i} - D(R_i + L_{(j+1),i}) \]
\[ I_{b(j,i)} = S_{j,i} - D(L_{j,i}) \]

Thus the net inventory at the beginning of the cycle \( I_{e(j,i)} \) is the inventory level just after the production order \((j, i)\) has been finished and the net inventory at the end of the cycle \( I_{b(j,i)} \) is the inventory level just before the following order \( j + 1 \) will be finished. Finally, the average inventory \( E[I_{j,i}] \) is needed for the evaluation of the schedule. We approximate this by the mean of the inventory level at the beginning and the end of a cycle (see van der Heijden et al. (1997)):

\[ E[I_{j,i}] = \frac{E[(I_{b(j,i)})^+] + E[(I_{e(j,i)})^+]}{2} \]  

(8)

Then, the average inventory of an item for the complete cycle is given by

\[ E[I_t] = \frac{\sum_{j=1}^{J_i} R_i E[I_{j,i}]}{T}. \]  

(9)

In the following Section will use this approximation to derive the objective function of the planning model and present a Local Search algorithm to optimize the schedule.

### 3 Optimization

The optimization algorithm minimizes the sum of the long-run expected setup and inventory costs while satisfying customer service requirements defined by a given fill rate. The result of the optimization is a detailed schedule, which controls the order generation process and is repeated infinitely. We code the solution based on the basic period approach, which means that the individual product cycles are multiples of some basic period. The algorithm consists of three parts: The first one generates an initial solution based on a relaxed formulation of the problem. From this solution the second part searches iteratively for better solutions using a Local Search approach. For each new solution the feasibility has to be checked and the schedule has to be evaluated. This is done by the third part of the algorithm, a fast, myopic scheduling heuristic.
The total cost function of the SELSP is given by:

\[
TC = \sum_{i=1}^{N} \left( \frac{c_i}{T_{BP} k_i} + h_i \cdot E[\bar{I}_i] \right) \tag{10}
\]

The first part of the function calculates the setup costs per period \(c_i\) are the setups costs for item \(i\)\) and the second one the costs for holding inventory \(h_i\) denotes the holding costs per period and item). The objective function is different from the deterministic ELSP formulation, because we don’t consider continuous production availability but batch availability at the end of the production run. The average inventory \(E[\bar{I}_i]\) can be computed from equation (9), which is based on a pre-calculated schedule.

Our solution approach uses a Local Search mechanism to generate new solutions’ and to evaluate the corresponding objective function value \(TC\). In contrast to the deterministic version of the problem the solution to the SELSP cannot be described sufficiently by the cycle lengths of the individual products, because the order-up-to level (and hence also the average inventory) depends on the planned position and timing of each production interval in the cycle. Therefore, the solution to the SELSP can be described by the basic period \(T_{BP}\), the multipliers \(k_i\) and the time of the first setup of each product in the cycle \(t_{i0}\). Given the values for these variables the schedule is fixed by planned start and end time of each production run. Considering the fact that the construction of a feasible schedule is NP-hard (Hsu (1983)), we restrict the range of values for \(k_i\) to power-of-two (POT) values. This constraint facilitates the scheduling problem (but it doesn’t guarantee the feasibility of the scheduling problem) and causes only limited additional costs (see e.g. Haessler (1979)).

Our Local Search routine starts from an initial solution created by a simplified version of the Haessler&Hogue-heuristic (H&H-heuristic, Haessler and Hogue (1976)) for the deterministic relaxation of the SELSP. In this case the demand rate \(d_i[1]\) is approximated by the expected demand per period \(E[D(1)]\), which can be calculated from asymptotic relations derived by Cox (1962) (see also Smits and Wagner (2002)). The main steps of the algorithm are summarized in the following:

Then the H&H-solution is evaluated by the cost function \(TC\) which includes the average stock levels for the stochastic case. The Local Search algorithm used to
Algorithm 1 Simplified H&H-heuristic

[Step 1:] Calculate independent product cycles $t_i^{IS}$ (independent solution)
[Step 2:] Derive the multiples $k_i$ and the basic period $T^{BP}$ from $t_i^{IS}$
[Step 3:] Try to create a feasible schedule (see Algorithm 3 on page 15)
[Step 4:] If a feasible schedule is found, then stop, else extend the basic period and go to Step 3

improve the objective function value $C^0$ of the initial solution $z^0$ is based on Simulated Annealing (SA, see Kirkpatrick et al. (1983)). The procedure generates new solutions (neighbors) by perturbations of the current solution. Solutions with higher total costs $TC$ are accepted randomly with a certain probability, which decreases during the optimization.

In our algorithm the neighbors are generated by changing one of the values of the solution vector $\tilde{z} = \{T^{BP}, k_1, k_2, \ldots, k_N\}$. The variable for the first setup time $t_i^0$ is not part of the neighborhood as it is an implicit result of the feasibility check. The values of the solution vector are constraint by lower and upper bounds, which we derive from different properties of the deterministic solution:

- **Lower bound for $T^{BP}$**: We define the lower bound for the basic period for a given set of multipliers $k_i$. This dynamic approach allows us to increase the probability of selecting a feasible candidate during the optimization. The bound ensures, that all products can be produced according to their specific cycle, if the total production time can be balanced among all basic periods of the total cycle $T$. Therefore, the condition is a necessary but not sufficient constraint for feasibility.

$$T^{LB}_{BP} = \sum_i \frac{1}{k_i} \left( \frac{d_i[T^{BP}_L k_i]}{\rho_i} + s_i \right)$$

The expression $d_i[T^{BP}_L k_i]$ denotes the expected demand in an interval of length $T^{BP}_L \cdot k_i$.

- **Upper bound $T^{BP}$**: A simple solution to the ELSP can be generated if
the cycle length for all products is the same. This approach is referred to as the common-cycle approach and was introduced by Hanssmann (1962). The common cycle length is an upper bound on the cycle length in the best solution for the ELSP. If the demand is stochastic then additional stocks have to be kept (safety stocks). The cycles will be shorter in the stochastic solution than in the deterministic one, because this will improve the objective function \( TC \). Therefore, the common cycle length is also an upper bound for the SELSP.

We calculate the upper bound according to the optimal solution of Maxwell (1964), which we modified for the case of *batch availability* of production lots:

\[
T_{BP}^U = \max \left\{ \sqrt{\frac{2 \sum_i c_i}{\sum_i h_i \cdot d_i[1] \cdot 1 - \sum_i d_i[1]/\rho_i}} \right\}
\]

- **Lower bound for** \( k_i \): For each product we set the minimum cycle length to \( T_{BP} \), which implies, that all multiples are set equal to 1:

\[
k_i^L = 1
\]

- **Upper bound for** \( k_i \): For the deterministic ELSP Khouja et al. (1998) used an upper bound which is based on the so-called independent solution (IS). The IS is given by the individual Economic Order Quantities (EOQ) of each item \( i \). The corresponding individual cycle time \( t_i^{IS} \) is a lower bound on the optimal deterministic cycle length. Khouja et al. (1998) argue, that the cycle times in an ELSP-solution are close to the IS-solution, if the utilization \( \sum_i 1/\rho_i \) is low. For high utilization problems the cycles have to be increased to get a feasible solution. We also use the upper bound of Khouja et al. (1998) for the SELSP, since stochastic cycles tend to be shorter than the deterministic ones.

\[
k_i^U = \left[ \frac{t_i^{IS}}{T_{BP}} \cdot 5 \cdot \sum_i d_i[1]/\rho_i \right]
\]

The bounds defined above restrict the search to a set of reasonable solution vectors \( \hat{z} \). These vectors are generated by a neighborhood operation (perturbation), which incrementally changes the current solution. The neighborhood operation for the multipliers \( k_i^\lambda \) in solution \( \lambda \) is given by a step to the next lower or higher POT-value

\[
k_i^{\lambda+1} = \begin{cases} k_i^\lambda \cdot 2 & \text{if } i < \nu \leq i + 1 \\ k_i^\lambda / 2 & \text{if } i + N < \nu \leq i + N + 1, \end{cases}
\]

(11)
where $\nu$ is a uniformly distributed random number in $(1, 2N + 1]$. If the new multiple $k^{\lambda+1}_i \notin \{k^L_i, k^U_i\}$, then the original vector is restored and a new random number is drawn.

The neighborhood operation for a continuous variable in the Local Search context is only rarely addressed (see e.g. Ali et al. (2002)). Some authors propose two different operations, which are selected randomly. One is to go to a local neighboring value (small step), which improves the objective. The other is to use a uniform distribution on the domain of the variable and to select the new value randomly from this distribution. In our case we also use a two-way approach, which combines Local Search and larger steps in the solution domain. The first operation VOP1 varies the basic period $T^{BP}$ in $+/ - 1\%$ steps and the second one (VOP2) overcomes local optima by a step size of 50%.

The complete procedure for generating a new candidate $\hat{z}_{\lambda+1}$ is shown in Algorithm 2. The length of the solution vector is dependent on the number of items considered in the problem. Thus, the total number of potential neighborhood operations is $2N + 4$. To avoid problem size dependency of the solution process we choose a neighborhood operation for the multiples $k_i$ with the same probability as for the basic period $T^{BP}$. If an operation exceeds the domain defined by the bounds, then the routine is repeated until a feasible operation can be selected. Before this

**Algorithm 2 Generating a new candidate for the SELSP**

\[
\lambda := \text{current candidate (solution)} \\
\text{repeat} \\
\quad \nu = \text{uniformly distributed random variable } \in (0, 1] \\
\quad \text{if } \nu < 0.5 \\
\quad \quad \text{alter the basic period } T^{BP}_\lambda \text{ by applying VOP1 } +/ - \text{ or VOP2 } +/ - \\
\quad \text{else} \\
\quad \quad \text{apply equation (11) to change one of the multipliers } k^\lambda_i \\
\quad \text{until no bounds are exceeded } (k^L_i \leq k^{\lambda+1}_i \leq k^U_i, \forall i \text{ and } T^{BP}_L \leq T^{BP}_{\lambda+1} \leq T^{BP}_U) 
\]
new candidate $\hat{z}_{\lambda+1}$ can be evaluated a feasible schedule needs to be created. Since all product cycles are power-of-two multiples of a basic period, a simple one-pass heuristic performs satisfactory. The heuristic is based on an algorithm proposed by Tunasar and Rajgopal (1996) for the deterministic ELSP. The procedure is summarized in Algorithm 3. The heuristic defines time buckets and assigns items to

**Algorithm 3 Generating a schedule for the SELSP**

- Calculate the least common multiple $k_c$ of $k_1, k_2, \ldots, k_N$
- and the complete rotation cycle $T = T^{BP} \cdot k_c$
- Define time buckets $\omega_1, \ldots, \omega_{k_c}$ of length $T^{BP}$
- Sort all items in ascending order of their multiples $k_i$
- Sort all items with equal multiples in descending order of capacity usage
- **repeat**
  - Select the next item $i$ from the list
  - From all feasible time buckets select the one with the most capacity remaining
  - Assign item $i$ to this bucket, update the capacity usage and fix the first setup time $t_i^0$ for item $i$
  - Assign item $i$ to $k_c/k_i - 1$ subsequent time buckets with intervening gaps of $k_i - 1$ buckets and update the capacity usage
  - Determine the maximum capacity usage $c_{ap}^{\text{max}}$ of all time buckets
- **until** ($c_{ap}^{\text{max}} > T^{BP}$) or ($iteration = N$)
  - **if** $c_{ap}^{\text{max}} > T^{BP}$ then no feasible schedule was found

these buckets in ascending order of their multiples $k_i$. This is reasonable because the flexibility for scheduling is increasing for items which have to be produced less often in the total cycle $T$. Similarly, the flexibility for items with lower capacity usage is higher and thus, items with the same multiplier are sorted in descending
order of capacity usage. Then, the products are assigned to the time bucket with the most capacity remaining, in contrast to Tunasar and Rajgopal (1996), who assign the item to the first feasible time bucket found by the algorithm. The heuristic returns two important results for the optimization: First, it states the feasibility of a given solution and second, it returns the detailed timing of the production intervals by scheduling the first setup of each item $t_i^0$. Our tests on a large variety of different ELSP and SELSP problems showed, that in most cases the procedure finds a feasible schedule if one exists. Nevertheless, it should be noted that the heuristic cuts off some solutions as "infeasible" which are feasible.

4 Numerical results

In the following we will compare our optimization algorithm with other heuristics developed for the deterministic ELSP and give some results on the performance of our approach in a stochastic setting. All procedures were implemented in Borland Delphi 5.0 and the computations were run on a Windows 2000 PC with a Pentium III 850 Mhz processor. The parameters for the Simulated Annealing algorithm were chosen after running some initial performance tests and were fixed for the whole experiment. We used a standard cooling schedule which determines the temperature in iteration $\tau$ from the initial temperature $e_1$ and the cooling parameter $\alpha$ by $e_\tau = \alpha^\tau e_1$. In each iteration we evaluate $\tau$ candidates (plateau length). The parameters used in all tests are summarized in Table 2.

<table>
<thead>
<tr>
<th>Table 2: Simulated annealing parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>initial temperature $e_1$</td>
</tr>
<tr>
<td>cooling parameter $\alpha$</td>
</tr>
<tr>
<td>plateau length $\tau$</td>
</tr>
</tbody>
</table>

4.1 Deterministic problems

In contrast to the SELSP the ELSP is a standardized problem and therefore, a range of benchmark problems has been developed over time. All SELSP models found in literature are somehow different and cannot be compared to each other.
Most differences concern the way stochastic disruptions of the schedule are treated and how the demand process is modeled.

Therefore, we tested our optimization approach on deterministic benchmark problems to get an impression of the performance of the algorithm. We consider nine test problems from Bomberger (1966) (Bom1,...,Bom4) and Fujita (1978) (Fujita2,...,Fujita6) each with 10 items and compare the results to those of Tunasar and Rajgopal (1996). They also use the basic period approach with power-of-two multiples and solve the model by an evolutionary algorithm. We modified our heuristic in two ways to be able solve the deterministic ELSP appropriately: First, we changed the objective function to consider continuous availability of the production quantity and second, we added a routine to calculate the optimal basic period for a given set of multipliers. Therefore, the solution vector comprises only the multiples $k_i$.

In Table 3 the results are summarized and compared those of Tunasar and Rajgopal (1996). For each problem the first line shows the result of Tunasar and Rajgopal (1996), whereas the second and third one show the performance of our Local Search procedure with different plateau length. In most cases the algorithm found the same solution as reported by Tunasar and Rajgopal (1996) after only 300 candidate tests (plateau length 50). We examined also the robustness of the solution quality by different seed values for the initialization of the random variables used. Considering our results for the 10 products problems the algorithm seems to give robust solutions, because we got the same solution for 7 of the 9 test instances for 10 different seed values for the case of a plateau length of 500.

4.2 Stochastic problems

In contrast to the deterministic optimization the stochastic version requires much more computational time because of the additional approximation procedures for the lead time. Therefore, we examine rather small 4 product problems to get some insights into the solution properties of the SELSP. We generated 10 test problems with different cost structure but the same demand processes. The squared coefficient of variation of both, the arrival rate $A_i$ and the order size $D_i$, are set to 1. Furthermore, the service level is given by a fill rate of 95% for all four products.
Table 3: Results for the deterministic benchmark problems

<table>
<thead>
<tr>
<th>Problem</th>
<th>Plateau length</th>
<th>Time [min:sec]</th>
<th>TC</th>
<th>Basic period $T^{BP}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bom1</td>
<td>50</td>
<td>&lt; 00:01</td>
<td>4130.8</td>
<td>31.7</td>
</tr>
<tr>
<td></td>
<td>500</td>
<td>00:03</td>
<td>4148.7</td>
<td>43.5</td>
</tr>
<tr>
<td>Bom2</td>
<td>50</td>
<td>&lt; 00:01</td>
<td>6104.5</td>
<td>25.6</td>
</tr>
<tr>
<td></td>
<td>500</td>
<td>00:04</td>
<td>5729.2</td>
<td>31.5</td>
</tr>
<tr>
<td>Bom3</td>
<td>50</td>
<td>00:01</td>
<td>6843.6</td>
<td>26.4</td>
</tr>
<tr>
<td></td>
<td>500</td>
<td>00:03</td>
<td>6843.6</td>
<td>26.3</td>
</tr>
<tr>
<td>Bom4</td>
<td>50</td>
<td>00:01</td>
<td>7697.1</td>
<td>23.4</td>
</tr>
<tr>
<td></td>
<td>500</td>
<td>00:04</td>
<td>7697.1</td>
<td>23.4</td>
</tr>
<tr>
<td>Fujita2</td>
<td>50</td>
<td>&lt; 00:01</td>
<td>4730.6</td>
<td>15.4</td>
</tr>
<tr>
<td></td>
<td>500</td>
<td>00:03</td>
<td>4782.6</td>
<td>18.7</td>
</tr>
<tr>
<td>Fujita3</td>
<td>50</td>
<td>&lt; 00:01</td>
<td>8800.8</td>
<td>11.0</td>
</tr>
<tr>
<td></td>
<td>500</td>
<td>00:05</td>
<td>8806.8</td>
<td>10.9</td>
</tr>
<tr>
<td>Fujita4</td>
<td>50</td>
<td>00:01</td>
<td>21716.8</td>
<td>17.3</td>
</tr>
<tr>
<td></td>
<td>500</td>
<td>00:04</td>
<td>21716.8</td>
<td>17.3</td>
</tr>
<tr>
<td>Fujita5</td>
<td>50</td>
<td>00:01</td>
<td>4194.4</td>
<td>19.0</td>
</tr>
<tr>
<td></td>
<td>500</td>
<td>00:04</td>
<td>4210.7</td>
<td>17.8</td>
</tr>
<tr>
<td>Fujita6</td>
<td>50</td>
<td>00:01</td>
<td>21519.1</td>
<td>14.9</td>
</tr>
<tr>
<td></td>
<td>500</td>
<td>00:04</td>
<td>21519.1</td>
<td>14.9</td>
</tr>
</tbody>
</table>

First, we have a look at the solution quality and how it develops during the optimization process. For this, each problem is solved using 10 different seed values for the random variables and the coefficient of variation of the objective function $c_{TC}$ value in the best solution is calculated. The results given in Table 4 show, that the solution converges already after some iterations. This may be caused by the fact, that the stochastic solution is somewhere in the close neighborhood of the deterministic relaxation of the problem which is used for the initialization of the SELSP heuristic.

Next, we compare the results for different levels of uncertainty. The test problems
Table 4: Robustness of the SELSP-heuristic

<table>
<thead>
<tr>
<th>plateau length</th>
<th>10</th>
<th>20</th>
<th>30</th>
<th>40</th>
<th>50</th>
</tr>
</thead>
<tbody>
<tr>
<td>average $c_{TC}$</td>
<td>0.120</td>
<td>0.027</td>
<td>0.004</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>average CPU-time [min:sec]</td>
<td>00:18</td>
<td>00:38</td>
<td>00:53</td>
<td>01:06</td>
<td>01:13</td>
</tr>
</tbody>
</table>

are the same as above, except that the squared coefficient of variation for the order size $c_D^2$ is varied from 0.75 to 5. Each problem is run with 10 different seed values and the plateau length is set to 50. We measured the cost relative to the objective function value of the problem with $c_D^2 = 1$ (see Table 5). The results show quite low sensitivity of the costs concerning the demand variation, but they also underline the necessity for an integrated solution approach to the SELSP.

Table 5: Demand variation sensitivity of the SELSP

<table>
<thead>
<tr>
<th>$c_D^2$</th>
<th>0.50</th>
<th>0.75</th>
<th>1.00</th>
<th>1.25</th>
<th>1.50</th>
<th>1.75</th>
<th>2.00</th>
<th>3.00</th>
<th>5.00</th>
</tr>
</thead>
<tbody>
<tr>
<td>relative cost</td>
<td>95.6</td>
<td>97.8</td>
<td>100.0</td>
<td>102.2</td>
<td>104.3</td>
<td>106.4</td>
<td>108.4</td>
<td>116.5</td>
<td>132.8</td>
</tr>
</tbody>
</table>

5 Conclusions

This paper investigates the stochastic ELSP and proposes an integrative approach for the optimization of the schedule. Based on an analytical model for the calculation of the average inventory level of Smits and Wagner (2002) we develop an algorithm using Local Search as optimization technique. The algorithm includes a greedy heuristic to construct a cyclic schedule based on the basic period approach.

The proposed approach makes it possible to include the consequences of stochastic demands in the optimization explicitly. Most models found in literature first generate a cyclic schedule based on deterministic demand, combine it with a recovery policy for the stochastic case and afterwards analyze the system. Our heuristic was tested on deterministic and stochastic problems and performed well in both cases.

Future research may extend the model to a more flexible scheduling model (e.g. without power-of-two restriction) and compare our results to the dynamic scheduling approach. This would yield more flexibility than the cyclic approach to react to demand changes. Furthermore, the model may be modified to different demand processes to be comparable to other modeling approaches from literature (e.g. the
approach of Federgruen and Katalan (1998)).

References


Maxwell, W.L. (1964) The scheduling of economic lot sizes, Naval Research Logistics Quarterly, vol 11, p 89-124


