FEP WORKING PAPERS
FEP WORKING PAPERS

Research Work in Progress

n. 341, Nov. 2009

# A Genetic Algorithm for Lot Size and Scheduling under Capacity Constraints and Allowing Backorders

José Fernando Gonçalves
Paulo S. A. Sousa

LIAAD
Faculdade de Economia, Universidade do Porto

U.PORTO
FEP FACULDADE DE ECONOMIA
UNIVERSIDADE DO PORTO

# A Genetic Algorithm for Lot Size and Scheduling under Capacity Constraints and Allowing Backorders*

**José Fernando Gonçalves**

**Paulo S. A. Sousa**

**Faculdade de Economia do Porto, LIAAD**
**Rua Dr. Roberto Frias, s/n, 4200-464 Porto, Portugal**
**jfgoncal@fep.up.pt, paulus@fep.up.pt**

**Abstract**

This paper addresses the problem of scheduling economic lots in a multi-product single-machine environment. A mixed integer non-linear programming formulation is developed which finds the optimal sequence and economic lots. The model takes explicit account of initial inventories, setup times, allows setups to be scheduled at arbitrary epochs in continuous time and models backorders. To solve the problem we develop a hybrid approach, combining a genetic algorithm and linear programming. The approach is tested on a set of instances taken from the literature and compared with other approaches. The experimental results validate the quality of the solutions and the effectiveness of the proposed approach.

**Keywords**: ELSP, Lot-sizing, Control, Production, Scheduling, Optimization, Genetic algorithm.

## 1 Introduction

This paper considers the economic lot scheduling problem (ELSP). The ELSP considers the production and scheduling of multiple products on a single machine to minimize the total cost. The problem is one that occurs often in practice. Examples range from bottling of detergents or paint to plastic wrap of potato chips. The ELSP has the following features:

1. The machine can only produce one product at a time; .
2. The demand and production rates of each product are deterministic and constant;
3. The setup cost and setup times are independent of the production sequence;
4. The production facility has enough capacity to satisfy the predicted demand during the planning horizon;
5. The inventory holding cost is proportional to the level of inventory

This situation leads to the question of how one should schedule the products on this single machine. The issue is one of determining both a sequence in which the products will be produced and a lot size for each production run of each product. The question of a lot size arises because the system usually incurs a setup cost and a setup time when the machine switches from one product to the next. The cost may be due to cleaning or to scrap losses incurred when the machine is adjusted to process the next product. The setup times imply a down time during which the machine cannot produce, which, in turn, implies a need to carry inventory.

Since the ELSP is proven to be NP-hard, the research efforts have been focused on formulating near optimal cyclical schedules. Several heuristic solutions have been developed using one of the three following approaches: the *common cycle*, the *basic period*, and the *time-varying lot size*. The common cycle approach always obtains a feasible schedule and is the simplest to implement,

*Supported by Fundação para a Ciência e Tecnologia (FCT) project PTDC/GES/72244/2006.
Date: 2009-07-20. FEP working paper 2009.
Corresponding author: jfgoncal@fep.up.pt.

but in some cases, the solution is quite high in comparison to the lower bound (LB). The basic period approach allows different cycle times for different products, however imposes that the cycle times be an integer multiple of a basic period. The basic period approach generally produces a better solution to the ELSP than the common cycle approach, but obtaining a feasible schedule is NP-hard (Bomberger, 1966). Finally, the time-varying lot size approach provides more flexibility than the previous approaches, since it allows for different lot sizes for the different products within a cycle. Dobson (1987) shows that the time-varying lot size approach can always constructs better quality solutions and produce feasible schedules.

In this paper we develop a model that retains the assumptions of the ELSP formulation, but includes other appealing features. We require the solution to be consistent with given initial and ending inventory levels as well as to be capacity-feasible. Moreover, the model allows backorders and admits schedules out to a given planning horizon that need not be cyclical.

In Section 2 we present a brief review of academic literature relevant to the problem discussed above. In Section 3, a new mathematical programming formulation for the problem discussed above is presented. In section 4 we present a solution methodology for the mathematical formulation. In Section 5 we report the results of computational experiments. Section 6 presents some conclusions and suggests directions for further research.

## 2 Literature review

The economic lot scheduling problem has been studied extensively over the last 50 years. The first works comprise Eilon (1959), Rogers (1958), Hanssmann (1962), and Maxwell (1964). In these works, a lower bound (LB) on the cost of any solution is easily obtained by neglecting the capacity constraint of the machine, and solving the lot sizing problem for each product independently, resulting in the economic manufacturing quantity (EMQ) for each product. A tight lower bound (TLB) was developed by Bomberger (1966), whose determination admits only the capacity constraint. The usual approach to the ELSP is to determine a cyclic, repeatable schedule while attempting to minimize setup plus inventory costs per unit time. Several researchers re-discovered this TLB (Moon et al., 2002a). The ELSP research has concentrated on cyclic schedules which satisfy the Zero-Switch-Rule (ZSR), meaning a product is produced only if its inventory wipes out.

A comprehensive bibliography through 1976 is given by Elmaghraby (1978). Other references on the classical ELSP problem include Axsater (1984), Boctor (1982), Delporte and Thomas (1977), Doll and Whybark (1973), Fujita (1978), Haessler (1979), Hsu (1983), Maxwell and Singh (1983), and Saipe (1977).

Less restricted variants of the ELSP have been followed by other authors. Heuristic solution techniques using the approach of a discrete-time grid for setups have been proposed by Dixon and Silver (1981), Dogramaci et al. (1981), Newson (1975) and Van Nunen and Wessels (1978) among others. McKinney (1980) developed an optimal branch and bound technique. Two papers that account for setups are Fujita (1978) and Dobson (1987). Leachman and Gascon (1988) develop an heuristic for the case of time-varying, stochastic demands. Carreno (1990) suggests a heuristic procedure which applies a single-machine common cycle approach to the case of multiple machines. Zipkin (1991) presents an improved computational procedure for Dobson's formulation that uses a parametric quadratic programming approach. Dobson (1992) introduces a heuristic solution procedure for the case of sequence dependent setups. Wagner and Davis (2002) also approach the ELSP with sequence dependent setup times. Cattrysse et al. (1993) present a dual ascent and column generation heuristic to solve the formulation of the discrete-time lot sizing problem with setups.Gallego (1993), Gallego and Joneja (1994) and Moon (1994) extend the traditional ELSP by considering issues associated with raw materials and reduced production rates. Gonçalves et al. (1994) develop an heuristic for the case of multiple identical-machine production systems with time-varying, stochastic demands. Hindi (1995) develops algorithms for capacitated, multi-product lot-sizing without setups. Gonçalves (1987), Gallego and Roundy (1992) and Gonçalves and Leachman (1998) allow backorders in the ELSP. Khouja et al. (1998) successfully use the Genetic algorithm (GA) to the ELSP under the basic period approach. Moon et al. (2002b) propose a hybrid GA to the ELSP applying the time-varying lot size approach that has surpassed the classical Dobson's (1987) heuristic (DH). Feldmann and Biskup (2003) and Raza and Akgunduz (2008) use a simulated annealing based meta-heuristic. Gaafar (2006) apply GA to dynamic lot sizing with batch ordering. Raza et al. (2006) propose a Tabu Search (TS) algorithm and Neighborhood Search heuristics (NSa , NSb ) to solve the same problem.

Increasingly, research has taken common parameters as decision variables in manufacturing systems modeling. Faaland et al. (2004) introduce a new modeling frame work for the ELSP that allows for lost sales leading to higher profits.

In this paper we develop a model that combines some features of the classical ELSP with the other features of the discrete time lot sizing models. Like discrete time models, we develop a schedule out to a planning horizon. But we depart from discrete-time models in that time is treated as a continuous variable, whereby any point in time is eligible as an epoch to initiate a product setup. Like ELSP models, we assume costs and demand are stationary, but we differ from most previous approaches to the classical ELSP in that we include backorder costs and directly construct integrated schedules for all products that are sequence feasible and capacity feasible.

# 3 The Model and its Application

This section presents a a mixed integer non-linear mathematical programming formulation of the multi-product single machine problem, which takes explicitly account of given initial and ending inventory levels, allows for backorders and produces capacity-feasible schedules out to a given planning horizon that need not be cyclical. The following notation is introduced.

**Indices:**

| | | |
|---|---|---|
| $i$ | = | product index; $I$ denotes the set of all products; |
| $n$ | = | production interval; $n = 1, 2, ..., N$, where $N$ is the total number of production intervals; |
| $T_n$ | = | time epoch corresponding to the beginning to the $n^{\text{th}}$ production interval. |

**Parameters:**

| | | |
|---|---|---|
| $a_i$ | = | setup cost of product $i$; |
| $d_i$ | = | demand rate of product $i$; |
| $b_i$ | = | backorder cost per unit for product $i$; |
| $h_i$ | = | holding cost per unit and per unit time for product $i$; |
| $p_i$ | = | production rate of product $i$; |
| $s_i$ | = | setup time for product $i$; |
| $M$ | = | arbitrary large number; |
| $I_{i,S}$ | = | inventory level of product $i$ at the start of the planning horizon; |
| $I_{i,E}$ | = | minimum inventory level of product $i$ at the end of the planning horizon; |
| $LBH$ | = | lower bound on the planning horizon; for the purposes of computing costs, a planning horizon larger than $LBH$ is endogenously determined by the model. |

**Variables:**

| | | |
|---|---|---|
| $B_{i,n}$ | = | backorder level of product $i$ at the end of the production run (if any) following the $n^{\text{th}}$ possible setup; |
| $L_{i,n}^{+}$ | = | inventory level of product $i$ at the end of the $n^{\text{th}}$ possible setup if positive; |
| $L_{i,n}^{-}$ | = | inventory level of product $i$ at the end of the $n^{\text{th}}$ possible setup if negative; |
| $U_{i,n}$ | = | inventory level of product $i$ at the end of the production run (if any) following the $n^{\text{th}}$ possible setup; |
| $T_n$ | = | time epoch corresponding to the beginning to the $n^{\text{th}}$ production interval $n = 1, 2, \cdots, N$. Setup in interval $n$ begins at time $T_n$; |
| $T_{N+1}$ | = | the time epoch at or after the end of the production run in interval $N$ at which minimum ending inventories are achieved. $T_{N+1}$ serves as the planning horizon for computing costs; |
| $Z_{i,n}$ | = | quantity of product $i$ produced during the $n^{\text{th}}$ possible production run; |
| $X_{i,n}$ | = | 1 if $Z_{i,n} > 0$; 0 otherwise. |

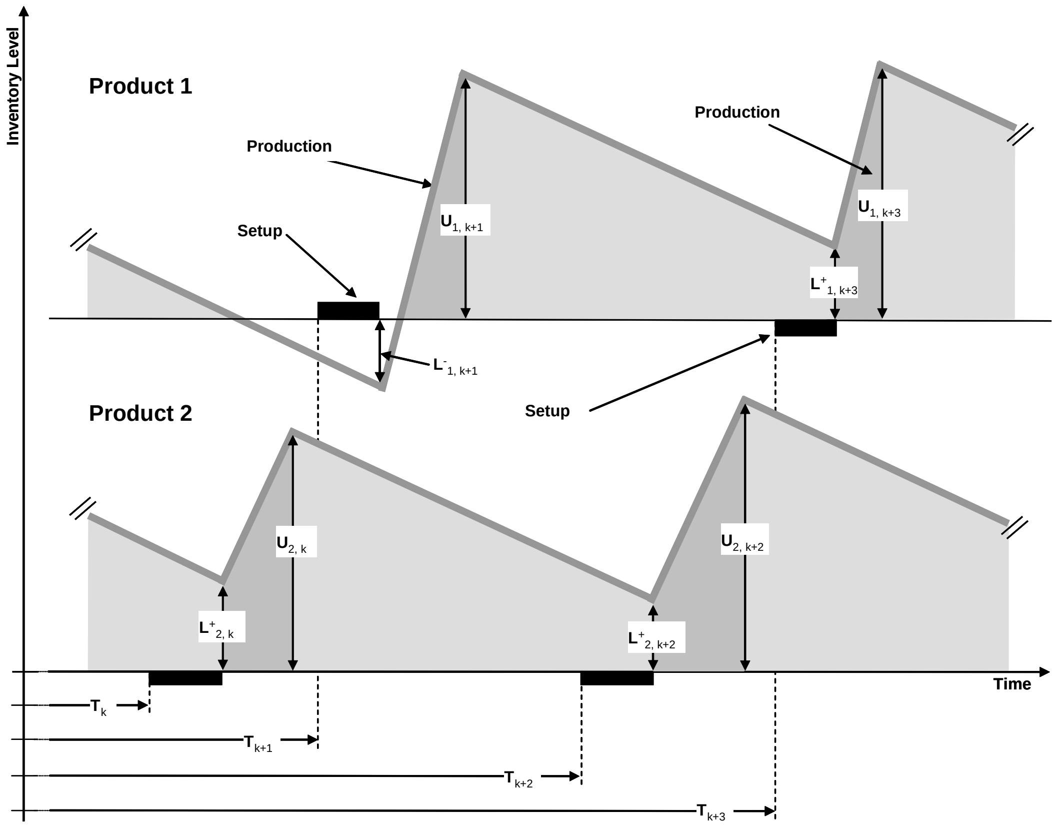


Figure 1: Defining Notation

Figure 1 illustrates the definition of the variables $L_{i,n}^{+}$, $L_{i,n}^{-}$, $T_n$, $U_{i,n}$. For all $i\,\epsilon\,I$ we make the following notational conventions: $L_{i,0}^{+}=I_{i,S}$, $U_{i,0}=I_{i,S}$ and $L_{i,N+1}^{+}=I_{i,E}$.The formulation of the scheduling problem is as follows. The objective function and the constraints will be explained below.

$$\text{Minimize}\;\; \frac{1}{T_{N+1}}\sum_{i\,\epsilon\,I}\sum_{n=0}^{n=N}\left\{a_iX_{i,n}+h_i\left[\frac{U_{i,n}^{2}-L_{i,n}^{+^{2}}}{2\,(p_i-d_i)}+\frac{U_{i,n}^{2}-L_{i,n+1}^{+^{2}}}{2\,d_i}\right]+b_iB_{i,n}^{2}\right\} \tag{1}$$

Subject to:

$$\sum_i X_{i,n}\leq 1 \quad n=1,...,N \tag{2}$$

$$Z_{i,n}\leq X_{i,n}M \qquad i\,\epsilon\,I;\quad n=1,...,N \tag{3}$$

$$T_n+\sum_i s_iX_{i,n}+\sum_i\frac{Z_{i,n}}{p_i}\leq T_{n+1}\quad n=1,...,N \tag{4}$$

$$I_{i,S}\;+\sum_{k=1}^{k=n-1}Z_{i,k}-d_i\left(T_n+\sum_j s_jX_{j,n}\right)=L_{i,n}^{+}-L_{i,n}^{-}\quad i\,\epsilon\,I;\;\; n=1,...,N \tag{5}$$

$$L_{i,n}^{+}-L_{i,n}^{-}+\left(1-\frac{d_i}{p_i}\right)Z_{i,n}\leq U_{i,n}\quad i\,\epsilon\,I;\quad n=1,...,N \tag{6}$$

$$L_{i,n}^{-}+(X_{i,n}-1)\,M\leq B_{i,n}\qquad i\,\epsilon\,I;\quad n=1,...,N \tag{7}$$

$$T_{N+1}\geq LBH \tag{8}$$

$$I_{i,S} + \sum_{k=1}^{k=N} Z_{i,k} - d_i T_{N+1} \geq I_{i,E} \qquad i \,\epsilon\, I \tag{9}$$

Variables:

$$B_{i,n},\, L^{+}_{i,n},\, L^{-}_{i,n},\, T_n,\, U_{i,n},\, Z_{i,n}, \geq 0;\;\; X_{i,n} \epsilon \{0,1\}\,, \quad i\,\epsilon\, I;\;\; n = 1, ..., N.$$

The objective function (1) represents the total cost per unit time during the interval [0, $T_{N+1}$]. The total cost of the solution proposed by the model is computed by adding, for all products and for all production intervals, the setup, the inventory costs and the backorder costs. The backorder costs are proportional to the square of the backorder level because it represents better the real life costs. The holding cost terms under the summation in (1) express the holding cost incurred for product $i$ during the intervals $[T_n + \sum_i s_i X_{i,n},\ T_{n+1} + \sum_i s_i X_{i,n+1}]$ for $n = 1, ..., N$, plus the holding cost during $[0,\ T_1 + \sum_i s_i X_{i,1}]$ in the $n = 0$ term and the holding cost during $[T_N + \sum_i s_i X_{i,N},\ T_{N+1}]$ in the $n = N$ term. As for the constraints, (2) states that at most one product can be produced in an production interval. Constraint (3) forces a product setup whenever there is production of that product. Constraint (4) ensures that the next production interval starts after the end of the production in the previous production interval. Constraint (4) also ensures that the production schedule obtained is capacity feasible. Constraint (5) measures the inventory level of a product at the end of every possible production setup, (see Fig. 1). Constraint (6) measures the inventory level of a product at the end of every possible production run (see Fig. 1). Constraint (7) measures the backorder level of a product at the end of the production intervals that have a production setup for that product. Constraint (8) guarantees that the length of the planning horizon has at least the predefined length $LBH$. Constraint (9) ensures that the inventory level of every product $i$ at the end of the planning horizon is greater or equal to a predefined level $I_{i,E}$.

The above model is to be applied using a rolling horizon of NRH time periods (e.g., eight weeks). The user would only implement the proposed schedule for the first time period in the horizon (e.g., one week). After that, the model would be re-run using a planning horizon which includes the most recent demand forecast for time periods 2 to NRH+1. This procedure would be repeated every time period.

# 4 Solution Methodology

## 4.1 Overview

The mixed integer non-linear formulation developed in section 3 is very difficult to solve. In order to obtain a solution, we developed a new solution methodology, which combines a random-keys based genetic algorithm with a surrogate linear programming formulation (see section 4.2) .

The role of the genetic algorithm (GA) is to evolve the encoded solutions, or *chromosomes*, which represent the production sequence. The linear programming formulation makes use of the production sequence supplied by the GA and provides a surrogate formulation for the original one which is easily solved by any linear programming software package.

For each chromosome, the following four phases are applied:

1. *Decoding of the production sequence.* The first phase is responsible for transforming the chromosome supplied by the genetic algorithm into a production sequence.

2. *Fixing of the binary variables in surrogate model.* In the second phase, the binary variables of the surrogate model are fixed according to the production sequence supplied by the GA in phase one.

3. *Solving the surrogate model.* The third phase consists in solving the surrogate linear programming formulation. This can be done by any standard linear programming software package.

4. *Fitness evaluation.* In the final phase, since the surrogate model does not provide the exact total cost for a given production sequence, we calculate the exact cost using the correct cost function and values of the variables given by the surrogate formulation. This exact total cost function is used as a fitness measure (quality measure) to feedback to the genetic algorithm.

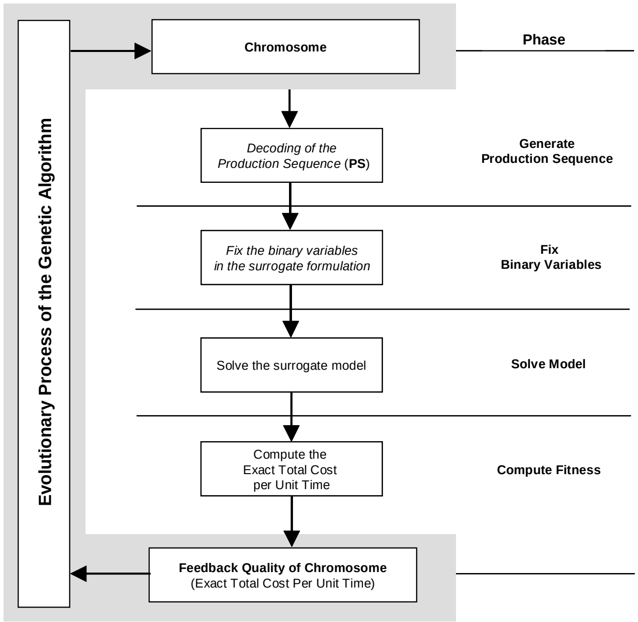


Figure 2: Architecture of the solution methodology.

Figure 2 illustrates the sequence of steps applied to each chromosome generated by the genetic algorithm.

The remainder of this section describes in detail the solution methodology.

## 4.2 Surrogate Model

The solution of the model developed in section 3 is not easy to obtain, given the fact that the objective function (1) is non-linear and is not positive definite. In order to overcome this problem, instead of using the exact inventory costs, we will use an upper bound on the inventory costs. The upper bound on the inventory costs is obtained by assuming that $L_{i,n}^{+^2} = L_{i,n}^{+^2} = 0$. This is the ideal case, realized if the setup is completed when the inventory level reaches zero. The factor $\frac{1}{T_{N+1}}$ is substituted by $\frac{1}{LBH}$, i.e., we assumed that constraint (8) is going to be tight. The modified objective function is:

$$\text{Minimize} \quad \frac{1}{LBH} \sum_{i \, \epsilon \, I} \sum_{n=0}^{n=N} \left\{ a_i X_{i,n} + h_i \left[ \frac{1}{2\,(p_i - d_i)} + \frac{1}{2\, d_i} \right] U_{i,n}^{^2} + b_i B_{i,n}^{^2} \right\} \qquad (10)$$

Note that even though this modified objective function does not compute the exact inventory costs, it still has the capability to drive the solution procedure to good if not optimal schedules.

In order to exclude from the objective function the terms in $U_{i,n}^2$ for the intervals $n$ in which there is no setup of product $i$ , constraint (6) will be redefined as follows:

$$L_{i,n}^{+} - L_{i,n}^{-} + \left(1 - \frac{d_i}{p_i}\right) Z_{i,n} + (X_{i,n} - 1)\, M \le U_{i,n} \qquad i \, \epsilon \, I; \quad n = 1, \ldots, N \qquad (11)$$

This way, $U_{i,n}^2$ will take the value zero if no setup for product $i$ is made in interval $n$. We refer to the model with (10) replacing (1) and (11) replacing (6) as the Surrogate Model. It is a mixed integer quadratic program with a positive-definite objective term, and it is thus more tractable than the model in Section 3.

The values of the binary variables in the above surrogate model are going to be fixed according to the production sequence supplied by the GA described in the next section. Therefore the above formulation becomes a formulation having only continuous variables with a positive-definite quadratic term in objective function and with linear constraints.

Instead of directly solving the quadratic mixed integer formulation for the surrogate model we approximated the quadratic objective (10) with a piece-wise linear curve. This way, basic integer linear programming (LP) optimization software available to us could be applied. We defined a sufficient number of linear segments so that the error in the objective function's holding cost introduced by the piece-wise linear approximation was not more than 5%. To solve the resulting LP, we used the open source CoinOR CLP software package Version 1.10.0.

After obtaining a solution to the surrogate model, we computed the exact total cost per unit time using the exact inventory costs and the exact value of $\frac{1}{T_{N+1}}$.

## 4.3 Genetic algorithm

The following sub-sections present the chromosome representation and decoding and the evolutionary process.

### 4.3.1 Chromosome representation and decoding

The genetic algorithm described in this paper uses a random-keys alphabet comprised of random real numbers between 0 and 1. The evolutionary strategy used is similar to the one proposed by Bean (1994), the main difference occurring in the crossover operator. The important feature of random keys is that all offspring formed by crossover are feasible solutions. This is accomplished by moving much of the feasibility issue into the objective function evaluation. If any random-key vector can be interpreted as a feasible solution, then any crossover vector is also feasible. Through the dynamics of the genetic algorithm, the system learns the relationship between random-key vectors and solutions with good objective function values.

The adequate representation of a solution plays a key role in the development of a genetic algorithm. A chromosome represents a solution to the problem and is encoded as a vector of random keys. Each solution chromosome is made of $N + 1$ genes, where $N$ is the maximum number of setups to be allowed in the production sequence. The first $N$ genes are used to obtain a *Maximal Production Sequence* (MPS), while $gene_{N+1}$ is used to obtain the number of setups in PS (NumSetup) , i.e.

$$Chromosome = (\underbrace{gene_1, \ldots, gene_N}_{\text{Production Sequence}}\ ,\quad \underbrace{gene_{N+1}}_{\text{Number of setups}}\ ).$$

Both the MPS and NumSetup are used to construct the production sequence (PS). The decoding (mapping) of the first $N$ genes of each chromosome into a MPS starts by constructing a vector with a initial sequence setups (IPS). Let $N_i$ represent an upper bound on the number of setups of product $i$ that are going to appear in MPS. The IPS can be constructed by filling an array of size $N = \sum_{i \in I} N_i$ with $N_i$ values equal to $i$ for each product $i$ (see Figure 3a). The value of NumSetup is decoded as follows:

$$NumSetup \ = \lceil MinSetup + gene_{N+1} \times (N - MinSetup) \rceil$$

where MinSetup is a lower bound on the number of setups in the optimal solution and $\lceil x \rceil =$ denotes the smallest integer greater than $x$.

The construction of a maximal production sequence (MPS) is accomplished by sorting the genes and the IPS in ascending order (see Figure 3b). Lastly, the PS is obtained by selecting only the first NumSetup setups in the MPS (see Figure 3c). Figure 3 depicts an example where $N_1 = 2,\ N_2 = 3,\ N_3 = 1,\ N_4 = 2,\ N_5 = 3$. The final production sequence is (5, 2, 4, 3, 1, 2, 5, 1).

Having obtained a PS, we can now proceed to fix the binary variables in the surrogate model. The fixing procedure is as follows

$$X_{i,n} = \begin{cases} 0 & \text{if } \ PS[n] = i \\ \\ 1 & \text{otherwise} \end{cases}$$

In some cases it is possible to obtain a MPS where a product appears consecutively in the sequence. In this case, only the first value of the consecutive sub-sequence is used. For example, assume that the following MPS was obtained (5, 2, 4, 4, 4, 3, 1, 2, 5, 1) and that NumSetup=6, then the PS would be (5, 2, 4, 3, 1, 2).

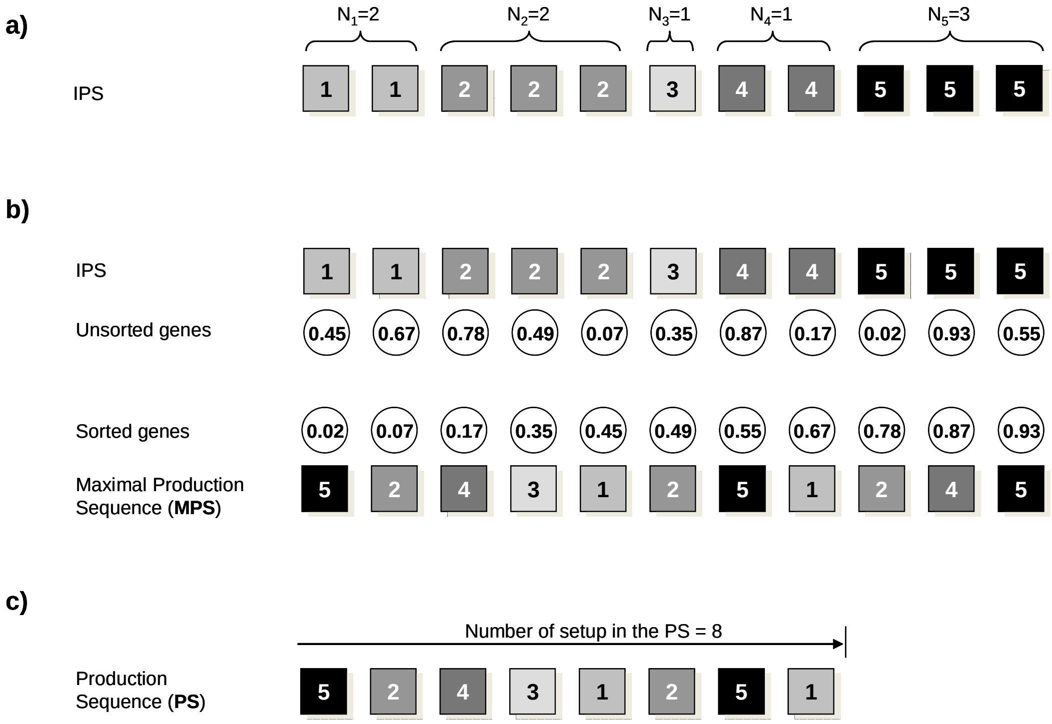


Figure 3: Chromosome decoding procedure.

#### 4.3.2 Evolutionary process

To breed good solutions, the random-key vector population is operated upon by a genetic algorithm. There are many variations of genetic algorithms obtained by altering the reproduction, crossover, and mutation operators. The reproduction and crossover operators determine which parents will have offspring, and how genetic material is exchanged between the parents to create those offspring. Mutation allows for random alteration of genetic material. Reproduction and crossover operators tend to increase the quality of the populations and force convergence. Mutation opposes convergence and replaces genetic material lost during reproduction and crossover.

The *population is initialized* with random-key vectors whose components are random real numbers uniformly sampled from the interval $[0, 1]$. *Reproduction* is accomplished by first copying some of the best individuals from one generation to the next, in what is called an *elitist strategy* (Goldberg, 1989). The advantage of an elitist strategy over traditional probabilistic reproduction is that the best solution is monotonically improving from one generation to the next. The potential downside is population convergence to a local minimum. This can, however, be overcome by an appropriate amount of mutation as described below.

*Parameterized uniform crossovers* (Spears and Dejong, 1991) are employed in place of the traditional one-point or two-point crossover. After two parents are chosen, one chosen randomly from the best (unlike Bean (1994), we always choose one parent from the best, Gonçalves and Resende (2009)show that this change produces results with better quality and faster) and the other chosen randomly from the full old population (including chromosomes copied to the next generation in the elitist selection), at each gene we toss a biased coin to select which parent will contribute the allele, *see TOP* in Figure 4. Figure 5 presents an example of the crossover operator. It assumes that a coin toss of heads selects the gene from the first parent, a tails chooses the gene from the second parent, and that the probability of tossing a heads, crossover probability $CProb = 0.7$. In Section 5 we describe how we determine this value empirically.

Rather than using the traditional gene-by-gene mutation with very small probability at each generation, some new members of the population are randomly generated from the same distribution as the initial population (see $BOT$ in Figure 4). The purpose of this process is to prevent premature convergence of the population, like in a mutation operator, and leads to a simple statement of convergence. Figure 4 depicts the transitional process between two consecutive generations.

The initial population is randomly generated.

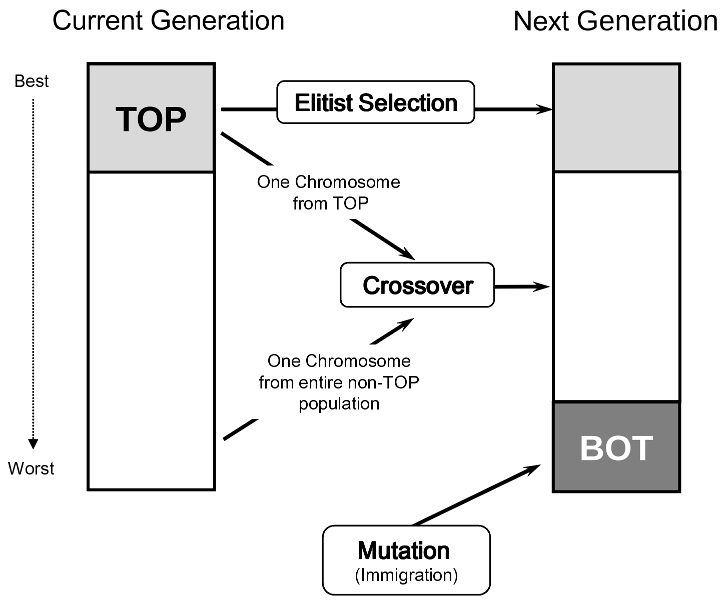


Figure 4: Transitional process between consecutive generations.

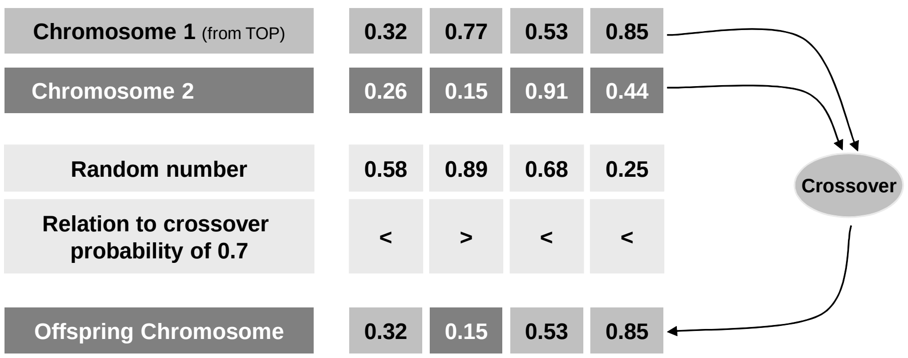


Figure 5: Example of parameterized uniform crossover with crossover probability equal to 0.7.

# 5 Numerical experiments

In this section we report the results obtained on a set of experiments conducted to evaluate the performance of **G**enetic **A**lgorithm for **L**ot **S**izing (GALS) proposed in this paper.

We have tested GALS in a real life industrial environment and the results were considered very good by the users. However, we cannot assess its quality since there are no benchmark instances for the real cases. In order to overcome this problem, we benchmark the quality of GALS against other approaches for the classical ELSP. To do that, we assume that we want to obtain a cyclical production sequence that minimizes the total setup and inventory holding cost.

To force our model to produce a cycle, we impose that the initial inventory be equal to the final inventory by considering $I_{i,S}$, $I_{i,E}$ as variables and making $I_{i,S} = I_{i,E}$ in the model. We use a large values for $b_i$in order to avoid backorders. LBH is made equal to the sum of the setups in PS.

## 5.1 Benchmark algorithms

We compare the GALS with the following approaches, which present the good computational results:

- *DH* - Dobson's Heuristic Dobson (1987).
- *GA* - Genetic Algorithm approach by Moon et al. (2002b).
- *TS* - Tabu Search approach by Raza et al. (2006).
- $NS_a$ , $NS_b$ - Neighborhood Search Heuristics by Raza et al. (2006).

Table 3: Range of Parameters in past implementations.

| Parameter | Interval |
|---|---|
| TOP | 0.10 - 0.25 |
| BOT | 0.15 - 030 |
| Crossover Probability (CProb) | 0.70 - 0.80 |

- *SA* - Simulated Annealing approach by Raza and Akgunduz (2008).

## 5.2 Test problem instances

The effectiveness of GALS is evaluated by solving the following two well-known problems in the ELSP literature:

Table 1: Benchmark Problems.

| Problem | Source | Size | Machine Load |
|---|---|---|---|
| Mallya | Mallya (1992) | 5 products | $k = 0.02$ |
| Bomberger | Bomberger (1966) | 10 products | $k = 0.01$ |

where $k = 1 - \sum_{i \in I} \frac{p_i}{d_i}$.

Further analysis was conducted by generating three sets, each with 50 randomly generated problems using the uniform distribution parameters given in table 2.

Table 2: Distribution for randomly generated data for the test problem.

| Parameters | Set 1 | Set 2 | Set 3 |
|---|---|---|---|
| Number of items (units) | $[5, 15]$ | $[5, 15]$ | $[5, 15]$ |
| Production rate (units/unit time) | $[2000, 20000]$ | $[4000, 20000]$ | $[1500, 30000]$ |
| Demand rate (units/unit time) | $[1500, 2000]$ | $[1000, 2000]$ | $[500, 2000]$ |
| Set-up time (time/unit) | $[1, 4]$ | $[1, 4]$ | $[1, 8]$ |
| Set-up cost ($) | $[50, 100]$ | $[50, 100]$ | $[10, 350]$ |
| Holding cost ($) | $[1/240, 6/240]$ | $[1/240, 5/240]$ | $[5/240000, 5/240]$ |
| $k$ | | $\leq 0.1$ | |

## 5.3 GA configuration

Configuring genetic algorithms is oftentimes more an art form than a science. In our past experience with genetic algorithms based on the same evolutionary strategy (see Gonçalves and Almeida (2002), Gonçalves and Resende (2004), Gonçalves et al. (2005) and Gonçalves (2007)), we obtained good results with values of TOP, BOT, and Crossover Probability (CProb) in the intervals shown in Table 3.

With this in mind and to obtain a reasonable configuration, we conducted a small pilot study including 15 randomly generated instances (5 from each of the 3 datasets presented in table 2). We tested all the combinations of the following values:

- $TOP \in \{0.10, 0.15, 0.20\}$;
- $BOT \in \{0.15, 0.20, 0.25\}$;
- $CProb \in \{0.70, 0.75, 0.80\}$;
- *Population size* =100.

The configuration that minimized the sum of the total cost per unit time, over the 15 pilot problem instances, was $TOP = 15\%$, $BOT = 10\%$, $CProb = 0.7$.

The configuration presented in Table 4 was held constant for all experiments and all problems instances.

The computational results presented in the next section demonstrate that this configuration not only provides excellent results in terms of solution quality but also is very robust.

Table 4: Configuration used on all runs in computational experiments.

| | |
|---|---|
| Population size | 100 |
| Crossover probability | 0.7 |
| TOP | The 15 % most fit chromosomes from the previous generation are copied to the next generation |
| BOT | The 10 % least fit chromosomes from the previous generation are replaced with randomly generated chromosomes |
| Fitness | Total cost per unit time |
| Stopping Criterion | Stop after 100 generations |

### 5.4 Computational results

Our algorithm (GALS) was implemented in C++ and the computational experiments were carried out on a computer with a Intel 2.66GHz Xeon Quadcore CPU with the Linux CentOS 5 operating system.

All the results are compared against a lower bound on the total cost named, Tight Lower Bound (TLB), and given by the solution of the following formulation:

$$\text{Minimize} \quad \sum_{i\in I}\left[\frac{a_i}{T_i} + \frac{1}{2}h_i d_i\left(1-\frac{d_i}{p_i}\right)T_i\right]$$

Subject to:

$$\sum_{i\in I}\frac{s_i}{T_i} \leq 1 - \sum_{i\in I}\frac{p_i}{d_i}$$

$$T_i \geq 0 \quad i \in I$$

Let $T_i^*$ be the optimal solution of the above formulation. The the upper bound on the number of setups of each product $i$ is given by,

$$N_i = \left\lceil \frac{max_j\left\{T_j^*\right\}}{T_i} \right\rceil + 2 \quad i \in I.$$

Recall that $N = \sum_{i\in I} N_i$. The lower bound in the total number of setups in PS is given by,

$$MinSetup = \sum_{i\in I} max\ \{1,\ N_i - 4\}.$$

Table 5 reports the comparison on Mallya´s and Bomberger problems. Tables 6 and 7 present the detailed solutions obtained by all the approaches on the Mallya and Bomberger's problems respectively ($f$ =production sequence and $t$ =production times).

On Mallya's problem, GALS obtained the best solution amongst all the other approaches. GALS obtained a value of 58.731 whereas the best of the other approaches obtained only 60.782, i.e., GALS improved the best known solution by 3.4%.

On Bomberger's problem, GALS produced two solutions of very good quality. The first one obtained the second best result (125.240) that is only 0.09% above the best reported solution (125.125) and uses a PS with 42 setups. The second solution produced by GALS obtained a value of 126.028, but uses a PS having only 21 setups. Contrarily to the other approaches, GALS does not use a predefined number of setups when looking for the optimal solution (only the minimum and maximum number of allowed setups in PS are predefined).

Tables 8, 10 and 12 use as performance measure the ratio of cost of each solution approach to the TLB cost. Tables 9, 11 and 13 use as performance measure the ratio between the cost of GALS and the cost each of the other solutions approaches. The results show that for dataset 1 GALS performs almost as well the best approaches. For datasets 2 and 3 (harder to solve than dataset1), the results show that GALS clearly outperforms all the other approaches.

Table 5: Comparison on Bomberger and Mallya's problem.

| Problem | TLB | Existing solutions | | | | | | GALS |
|---|---|---|---|---|---|---|---|---|
| | | DH | GA | TS | $NS_a$ | $NS_b$ | SA | |
| Mallya | 57.726 | 60.874 | 60.911 | 60.911 | 60.911 | 60.782 | 60.911 | 58.731 |
| Mallya* | – | – | – | 60.782 | – | – | 60.782 | |
| Bomberger | 122.945 | 128.339 | 126.12 | 125.31 | 125.754 | 130.346 | 125.135 | 125.240/126.028 |

*Both TS and SA algorithms use production frequencies rounded off to the power of 2.

Table 6: Production sequence found by each approach on Mallya's problem.

| Approach | Details |
|---|---|
| GALS | $f$ ={1, 3, 4, 2, 3, 1, 4, 3, 5} |
| | $t$ ={13.1348, 4.3078, 13.0320, 14.5449, 3.7594, 10.0511, 14.0696, 3.5550, 9.7431} |
| DH | $f$ ={3, 4, 3, 1, 2, 3, 4, 3, 1, 5} |
| | $t$ ={2.5047, 16.07, 4.4432, 12.774, 15.743, 2.0777, 13.263, 3.5533, 12.32, 10.546} |
| TS | $f$ ={2, 3, 5, 4, 1, 3, 2, 4, 3, 1, 4} |
| | $t$ ={9.192, 5.615, 12.919, 11.607, 11.647, 3.412, 10.093, 11.596, 6.382, 19.094, 12.730} |
| $NS_a$ | $f$ ={3, 1, 4, 2, 3, 5, 4, 1, 3, 2, 4} |
| | $t$ ={6.382, 19.094, 12.730, 9.192, 5.615, 12.919, 11.607, 11.647, 3.412, 10.093, 11.596} |
| $NS_b$ | $f$ ={2, 3, 1, 4, 3, 5, 1, 3, 4, 3} |
| | $t$ ={15.743, 3.8749, 10.551, 14.329, 3.8914, 10.546, 14.543, 2.3425, 15.004, 2.4701} |
| TS* | $f$ ={1, 4, 3, 5, 1, 3, 4, 3, 2, 3} |
| | $t$ ={10.5512, 14.3294, 3.89142, 10.546, 14.5431, 2.34248, 15.0036, 2.47009, 15.7427, 3.87493} |
| SA | $f$ ={4, 2, 3, 5, 4, 1, 3, 2, 4, 3, 1} |
| | $t$ ={12.7299, 9.19219, 5.61506, 12.9188, 11.6074, 11.6471, 3.41226, 10.0926, 11.5956, 6.38185, 19.0935} |
| SA* | $f$ ={5, 1, 3, 4, 3, 2, 3, 1, 4, 3} |
| | $t$ ={10.546, 14.543, 2.3425, 15.004, 2.4701, 15.743, 3.8749, 10.551, 14.329, 3.8914} |

Table 7: Production sequence found by each approach on Bomberger's problem.

| Approach | Details |
|---|---|
| GALS (125.240) | $f=${8, 5, 9, 8, 4, 3, 2, 8, 5, 4, 8, 9, 8, 10, 3, 5, 8, 4, 2, 6, 8, 7, 9, 8, 4, 5, 3, 8, 1, 2, 4, 8, 9, 5, 8, 4, 3, 10, 8, 2, 4, 6} |
| | $t=${37.2170, 12.4236, 75.4461, 42.7729, 38.0125, 35.4729, 28.5132, 41.4490, 13.8730, 84.8128, 31.8461, 75.4363, 34.6915, 25.9949, 39.7331, 16.4352, 42.1577, 68.8510, 19.8643, 11.8018, 38.2823, 17.6489, 72.2861, 37.0035, 37.5254, 13.8418, 36.3635, 41.3386, 23.5315, 21.2389, 53.8994, 38.4765, 76.8774, 14.0252, 39.5483, 36.0992, 37.0626, 21.0718, 36.8347, 18.6334, 57.3347, 11.7312} |
| GALS (126.028) | $f=${5, 8, 7, 9, 8, 4, 3, 2, 8, 1, 4, 5, 8, 10, 9, 8, 4, 3, 2, 8, 6} |
| | $t=${18.4208, 38.7088, 9.2385, 81.7235, 41.8399, 36.6488, 39.6468, 23.4563, 39.9284, 12.3176, 64.0482, 18.5332, 42.0842, 24.6347, 75.3302, 40.4700, 35.5690, 38.1493, 22.7349, 38.5912, 12.3169, 60.8182} |
| DH | $f=${8, 4, 5, 8, 9, 8, 4, 10, 6, 8, 3, 2, 1, 8, 4, 5, 8, 9, 8, 4, 10, 7, 8, 3, 2, 8, 4, 5, 8, 9, 8, 4, 10, 6, 8, 3, 2, 8, 4, 5, 8, 9, 8, 4, 10, 7, 8, 3, 2} |
| | $t=${35.8256, 62.313, 22.8308, 40.0922, 95.293, 37.4462, 61.6508, 13.9507, 13.5698, 40.1746, 42.6083, 25.2726, 27.8604, 30.4963, 53.124, 19.1867, 34.6237, 82.1248, 30.5236, 48.4096, 13.1926, 10.1493, 28.2923, 42.1434, 25.11, 32.3382, 56.4482, 20.298, 36.5595, 86.7862, 33.3795, 51.2624, 13.8256, 14.2906, 29.3761, 43.702, 26.1615, 33.6866, 58.7276, 21.2655, 38.316, 91.0158, 33.6701, 53.8305, 14.7519, 10.746, 31.6915, 47.5066, 27.9323} |
| TS | $f=${4, 8, 5, 9, 8, 6, 7, 3, 2, 8, 4, 5, 10, 8, 9, 8, 4, 3, 8, 5, 2, 1, 4, 8, 9, 8, 4, 6, 5, 3, 8, 2, 10, 4, 8, 9, 5, 8, 4, 3, 2, 8} |
| | $t=${93.1766, 36.0186, 14.2169, 70.7668, 39.1481, 12.2983, 17.6698, 36.6966, 25.4799, 39.3299, 57.1912, 14.1954, 22.0708, 31.5664, 74.7627, 36.6643, 46.959, 40.4543, 46.3969, 14.6485, 22.5563, 23.5598, 49.5853, 32.9021, 77.9791, 44.8312, 46.0915, 11.2615, 13.6737, 35.3029, 40.6085, 18.7135, 25.0488, 53.1489, 38.4438, 76.8786, 13.9449, 37.2705, 30.8039, 36.3449, 21.5995, 38.9538} |
| $NS_a$ | $f=${4, 2, 8, 5, 9, 6, 8, 4, 1, 3, 8, 2, 5, 4, 8, 9, 7, 8, 4, 3, 5, 2, 8, 10, 4, 6, 8, 9, 5, 8, 4, 3, 2, 8, 4, 8, 5, 9, 8, 10, 3, 8} |
| | $t=${68.1266, 20.3381, 44.3999, 14.2791, 80.086, 10.551, 44.1094, 43.7623, 23.5598, 37.8943, 39.9535, 20.8636, 15.1851, 58.9104, 41.2746, 79.4703, 17.6698, 45.9338, 39.5632, 36.6152, 12.0651, 20.8661, 36.4121, 18.1524, 55.52, 13.0088, 36.6449, 73.3055, 13.1861, 37.678, 29.8507, 33.5974, 26.2814, 33.9898, 81.2233, 35.398, 15.9639, 67.5254, 29.2912, 28.9671, 40.6917, 37.0489} |
| $NS_b$ | $f=${8, 4, 2, 8, 10, 3, 5, 8, 4, 7, 8, 9, 8, 1, 4, 2, 8, 5, 3, 6, 8, 10, 4, 2, 8, 9, 8, 4, 8, 3, 5, 8, 10, 4, 8, 9, 8, 2, 7, 4, 8, 6, 3, 8, 5, 4, 8, 9, 10} |
| | $t=${39.3604, 35.9223, 36.8779, 20.9805, 36.6372, 18.5442, 57.0423, 11.6366, 37.0452, 12.362,a 75.0937, 42.524, 37.8421, 35.3653, 28.1914, 41.3358, 13.8737, 84.5389, 31.5848, 74.8069, 35.1145, 26.1391, 40.3426, 16.6999, 40.7355, 64.6932, 20.4757, 11.9232, 38.9956, 17.6698, 73.9825, 39.305, 43.2727, 13.7869, 36.2127, 41.2029, 23.5598, 21.1378, 53.645, 38.2932, 76.504, 13.9568} |
| SA | $f=${8, 4, 3, 10, 8, 2, 4, 6, 8, 5, 9, 8, 4, 3, 2, 8, 5, 4, 8, 9, 8, 10, 3, 5, 8, 4, 2, 6, 8, 7, 9, 8, 4, 5, 3, 8, 1, 2, 4, 8, 9, 5} |
| | $t=${39.3604, 35.9223, 36.8779, 20.9805, 36.6372, 18.5442, 57.0423, 11.6366, 37.0452, 12.362,a 75.0937, 42.524, 37.8421, 35.3653, 28.1914, 41.3358, 13.8737, 84.5389, 31.5848, 74.8069, 35.1145, 26.1391, 40.3426, 16.6999, 40.7355, 64.6932, 20.4757, 11.9232, 38.9956, 17.6698, 73.9825, 39.305, 43.2727, 13.7869, 36.2127, 41.2029, 23.5598, 21.1378, 53.645, 38.2932, 76.504, 13.9568} |

Table 8: Comparison of approaches with TLB on randomly generated problems using Set 1.

| | Comparison with Tight Lower Bound (TLB) | | | | | | |
|---|---|---|---|---|---|---|---|
| | $\frac{DH}{TLB}$ | $\frac{SA}{TLB}$ | $\frac{TS}{TLB}$ | $\frac{NS_a}{TLB}$ | $\frac{NS_b}{TLB}$ | $\frac{GA}{TLB}$ | $\frac{GALS}{TLB}$ |
| Mean | 1.0517 | 1.0243 | 1.0242 | 1.0250 | 1.0484 | 1.0302 | 1.0297 |
| Min. | 1.0114 | 1.0074 | 1.0074 | 1.0074 | 1.0114 | 1.0122 | 1.0041 |
| Max. | 1.2216 | 1.1098 | 1.1100 | 1.1192 | 1.2176 | 1.0564 | 1.1014 |

Table 9: Comparison of approaches with GALS on randomly generated problems using Set 1.

| | Comparison with GALS | | | | | |
|---|---|---|---|---|---|---|
| | $\frac{GALS}{DH}$ | $\frac{GALS}{SA}$ | $\frac{GALS}{TS}$ | $\frac{GALS}{NS_a}$ | $\frac{GALS}{NS_b}$ | $\frac{GALS}{GA}$ |
| Mean | 0.9791 | 1.0053 | 1.0054 | 1.0046 | 0.9822 | 0.9996 |
| Min. | 0.9928 | 0.9967 | 0.9967 | 0.9967 | 0.9928 | 0.9920 |
| Max. | 0.9016 | 0.9925 | 0.9923 | 0.9841 | 0.9046 | 1.0426 |

Table 10: Comparison of approaches with TLB on randomly generated problems using Set 2.

| | Comparison with Tight Lower Bound (TLB) | | | | | |
|---|---|---|---|---|---|---|
| | $\frac{DH}{TLB}$ | $\frac{SA}{TLB}$ | $\frac{TS}{TLB}$ | $\frac{NS_a}{TLB}$ | $\frac{NS_b}{TLB}$ | $\frac{GALS}{TLB}$ |
| Mean | 1.0503 | 1.0274 | 1.0272 | 1.0278 | 1.0491 | 1.0272 |
| Min. | 1.0070 | 1.0051 | 1.0051 | 1.0051 | 1.0070 | 1.0046 |
| Max. | 1.2336 | 1.0713 | 1.0708 | 1.0748 | 1.2054 | 1.0596 |

Table 11: Comparison of approaches with GALS on randomly generated problems using Set 2.

| | Comparison with GALS | | | | |
|---|---|---|---|---|---|
| | $\frac{GALS}{DH}$ | $\frac{GALS}{SA}$ | $\frac{GALS}{TS}$ | $\frac{GALS}{NS_a}$ | $\frac{GALS}{NS_b}$ |
| Mean | 0.9780 | 0.9998 | 1.0000 | 0.9994 | 0.9791 |
| Min. | 0.9977 | 0.9995 | 0.9995 | 0.9995 | 0.9977 |
| Max. | 0.8589 | 0.9890 | 0.9895 | 0.9858 | 0.8790 |

Table 12: Comparison of approaches with TLB on randomly generated problems using Set 3.

| | Comparison with Tight Lower Bound (TLB) | | | | | |
|---|---|---|---|---|---|---|
| | $\frac{DH}{TLB}$ | $\frac{SA}{TLB}$ | $\frac{TS}{TLB}$ | $\frac{NS_a}{TLB}$ | $\frac{NS_b}{TLB}$ | $\frac{GALS}{TLB}$ |
| Mean | 1.2550 | 1.1594 | 1.1592 | 1.1745 | 1.2301 | 1.1217 |
| Min. | 1.0193 | 1.0111 | 1.0107 | 1.0123 | 1.0192 | 1.0194 |
| Max. | 8.1570 | 5.3715 | 5.3720 | 5.4625 | 7.3938 | 3.8446 |

Table 13: Comparison of approaches with GALS randomly generated problems using Set 3.

| | Comparison with GALS | | | | |
|---|---|---|---|---|---|
| | $\frac{GALS}{DH}$ | $\frac{GALS}{SA}$ | $\frac{GALS}{TS}$ | $\frac{GALS}{NS_a}$ | $\frac{GALS}{NS_b}$ |
| Mean | 0.8938 | 0.9675 | 0.9677 | 0.9550 | 0.9119 |
| Min. | 1.0001 | 1.0082 | 1.0086 | 1.0070 | 1.0002 |
| Max. | 0.4713 | 0.7157 | 0.7157 | 0.7038 | 0.5200 |

# 6 Concluding remarks

In this paper we addressed the problem of scheduling economic lots in a multi-product single-machine environment. A mixed integer non-linear programming formulation is developed which finds an the optimal sequence and optimal economic lots. The model takes explicit account of initial inventories, setup times and allows setups to be scheduled at arbitrary epochs in continuous time and allows backorders. To solve the problem we develop a hybrid approach combining a random key based genetic algorithm and linear programming. The approach was tested on several benchmark instances taken from the literature and compared with other five approaches. The experimental results demonstrate the effectiveness and robustness of the proposed approach when compared with other approaches.

Further research might be directed for the case where there are multiple identical machines.

## Acknowledgments

This work has been supported by funds granted by Fundação para a Ciência e Tecnologia (FCT) project PTDC/GES/72244/2006.

## References

Axsater, S. (1984). Lower bounds for the economic lot scheduling problem using aggregation. *European Journal of Operational Research*, 17(2):201–206.

Bean, J. C. (1994). Genetics and random keys for sequencing and optimization. *ORSA Journal on Computing*, 6:154–160.

Boctor, F. F. (1982). Two-product, single-machine, static demand, infinite horizon lot scheduling problem. *Management Science*, 28(7):798–807.

Bomberger, E. E. (1966). A dynamic programming approach to a lot size scheduling problem. *Management Science*, 12(11):778–784.

Carreno, J. J. (1990). Economic lot scheduling for multiple products on parallel identical processors. *Management Science*, 36(3):348–358.

Cattrysse, D., Salomon, M., Kuik, R., and Van Wassenhove, L. N. (1993). Dual ascent and column generation heuristic for the discrete lotsizing and scheduling problem with setup times. *Management Science*, 39(4):477–486.

Delporte, C. M. and Thomas, L. J. (1977). Lot sizing and sequencing for n products on one facility. *Management Science*, 23(10):1070–1079.

Dixon, P. S. and Silver, E. A. (1981). A heuristic solution procedure for the multi-item, single-level, limited capacity, lot-sizing problem. *Journal of Operations Management*, 2(1):23–39.

Dobson, G. (1987). Economic lot-scheduling problem: Achieving feasibility using time-varying lot sizes. *Operations Research*, 35(5):764–771.

Dobson, G. (1992). Cyclic lot scheduling problem with sequence-dependent setups. *Operations Research*, 40(4):736–749.

Dogramaci, A., Panayiotopoulos, J. C., and Adam, N. R. (1981). Dynamic lot-sizing problem for multiple items under limited capacity. *AIIE Transactions (American Institute of Industrial Engineers)*, 13(4):294–303.

Doll, C. L. and Whybark, D. C. (1973). Iterative procedure for the single-machine multi-product lot scheduling problem. *Management Science*, 20(1):50–55.

Eilon, S. (1959). Economic batch-size determination for multi-product scheduling. *Operations Research*, 10(4):217–227.

Elmaghraby, S. E. (1978). The economic lot scheduling problem (elsp): review and extensions. *Management Science*, 24(6):587–598.

Faaland, B., Schmitt, T., and Arreola-Risa, A. (2004). Economic lot scheduling with lost sales and setup times. *IIE Transactions*, 36(7):629–640.

Feldmann, M. and Biskup, D. (2003). Single-machine scheduling for minimizing earliness and tardiness penalties by meta-heuristic approaches. *Computers & Industrial Engineering*, 44(2):307–323.

Fujita, S. (1978). Application of marginal analysis to the economic lot scheduling problem. *AIIE Trans*, 10(4):354–361.

Gaafar, L. (2006). Applying genetic algorithms to dynamic lot sizing with batch ordering. *Computers & Industrial Engineering*, 51(3):433–444.

Gallego, G. (1993). Reduced production rates in the economic lot scheduling problem. *International Journal of Production Research*, 31(5):1035–1046.

Gallego, G. and Joneja, D. (1994). Economic lot scheduling problem with raw material considerations. *Operations Research*, 42(1):92–101.

Gallego, G. and Roundy, R. (1992). The economic lot scheduling problem with finite backorder costs. *Naval Research Logistics*, 39(5):729–739.

Goldberg, D. (1989). *Genetic algorithms in search optimization and machine learning.* Addison-Wesley.

Gonçalves, J. F. (1987). *Continuous time analysis of production planning and scheduling models.* PhD thesis, University of California, Berkeley.

Gonçalves, J. F. and Leachman, R. C. (1998). A hybrid heuristic and linear programming approach to multi-product machine scheduling. *European Journal of Operational Research*, 110(3):548–563.

Gonçalves, J. F., Leachman, R. C., Gascon, A., and Xiong, Z. K. (1994). A heuristic scheduling policy for multi-item, multi-machine production systems with time-varying, stochastic demands. *Management Science*, 40(11):1455–1468.

Gonçalves, J. F. (2007). A hybrid genetic algorithm-heuristic for a two-dimensional orthogonal packing problem. *European Journal of Operational Research*, 183:1212–1229.

Gonçalves, J. F. and Almeida, J. R. (2002). A hybrid genetic algorithm for assembly line balancing. *Journal of Heuristics*, 8:629–642.

Gonçalves, J. F., Mendes, J. J. M., and Resende, M. G. C. (2005). A hybrid genetic algorithm for the job shop scheduling problem. *European Journal of Operational Research*, 167:77–95.

Gonçalves, J. F. and Resende, M. G. C. (2004). An evolutionary algorithm for manufacturing cell formation. *Computers and Industrial Engineering*, 47:247–273.

Gonçalves, J. F. and Resende, M. G. C. (2009). Biased random key genetic algorithms for combinatorial optimization. Technical report, AT&T Labs Research Technical Report, Florham Park, NJ 07733 USA.

Haessler, R. W. (1979). Improved extended basic period procedure for solving the economic lot scheduling problem. *AIIE Transactions (American Institute of Industrial Engineers)*, 11(4):336–340.

Hanssmann, F. (1962). *Operations research in production and inventory control.* Wiley.

Hindi, K. S. (1995). Algorithms for capacitated, multi-item lot-sizing without set-ups. *Journal of the Operational Research Society*, 46:465–472.

Hsu, W. L. (1983). On the general feasibility test of scheduling lot sizes for several products on one machine. *Management Science*, 29(1):93–105.

Khouja, M., Michalewicz, Z., and Wilmot, M. (1998). The use of genetic algorithms to solve the economic lot size scheduling problem. *European Journal of Operational Research*, 110(3):509–524.

Leachman, R. C. and Gascon, A. (1988). A heuristic scheduling policy for multi-item, single-machine production systems with time-varying, stochastic demands. *Management Science*, 34:1201–1205.

Mallya, R. (1992). Multi-product scheduling on a single machine: a case study. *Omega*, 20:529–534.

Maxwell, W. L. (1964). The scheduling of economic lot sizes. *Naval Research Logistics Quarterly*, 11(2-3):89–124.

Maxwell, W. L. and Singh, H. (1983). Effect of restricting cycle times in the economic lot scheduling problem. *IIE Transactions (Institute of Industrial Engineers)*, 15(3):235–241.

McKinney, J. (1980). *Optimal Multi-product Scheduling on One Machine over a Finite Horizon.* PhD thesis, The John Hopkins University.

Moon, I. (1994). Multiproduct economic lot size models with investment costs for setup reduction and quality improvement: review and extensions. *International journal of production research*, 32(12):2795–2801.

Moon, I., Giri, B., and Choi, K. (2002a). Economic lot scheduling problem with imperfect production processes and setup times. *Journal of the Operational Research Society*, 53(6):620–629.

Moon, I., Silver, E. A., and Choi, S. (2002b). Hybrid genetic algorithm for the economic lot-scheduling problem. *International Journal of Production Research*, 40(4):809–824.

Newson, E. F. P. (1975). Multi-item lot size scheduling by heuristic - 1. with fixed resources. *Management Science*, 21(10):1186–1193.

Raza, A. S. and Akgunduz, A. (2008). A comparative study of heuristic algorithms on economic lot scheduling problem. *Computers & Industrial Engineering*, 55(1):94–109.

Raza, S. A., Akgunduz, A., and Chen, M. Y. (2006). A tabu search algorithm for solving economic lot scheduling problem. *Journal of Heuristics*, 12(6):413–426.

Rogers, J. (1958). A computational approach to the economic lot scheduling problem. *Management Science*, 4(3):264–291.

Saipe, A. L. (1977). Production runs for multiple products: The two-product heuristic. *Management Science*, 23(12):1321–1327.

Spears, W. M. and Dejong, K. A. (1991). On the virtues of parameterized uniform crossover. In *Proceedings of the Fourth International Conference on Genetic Algorithms*, pages 230–236.

Van Nunen, J. and Wessels, J. (1978). Multi-item lot size determination and scheduling under capacity constraints. *European Journal of Operational Research*, 2(1):36–41.

Wagner, B. J. and Davis, D. J. (2002). A search heuristic for the sequence-dependent economic lot scheduling problem. *European Journal of Operational Research*, 141(1):133–146.

Zipkin, P. H. (1991). Computing optimal lot sizes in the economic lot scheduling problem. *Operations Research*, 39(1):56–63.

# *Recent FEP Working Papers*

| | |
|---|---|
| Nº 340 | Nuno Gonçalves and Ana Paula Africano, "*The Immigration and Trade Link in the European Union Integration Process*", November 2009 |
| Nº 339 | Filomena Garcia and Joana Resende, "*Conformity based behavior and the dynamics of price competition: a new rational for fashion shifts*", October 2009 |
| Nº 338 | Nuno Torres, Óscar Afonso and Isabel Soares, "*Natural resources, economic growth and institutions – a panel approach*", October 2009 |
| Nº 337 | Ana Pinto Borges, João Correia-da-Silva and Didier Laussel, "*Regulating a monopolist with unknown bureaucratic tendencies*", October 2009 |
| Nº 336 | Pedro Rui Mazeda Gil, "*Animal Spirits and the Composition of Innovation in a Lab-Equipment R&D Model*", September 2009 |
| Nº 335 | Cristina Santos and Aurora A.C. Teixeira, "*The evolution of the literature on entrepreneurship. Uncovering some under researched themes*", September 2009 |
| Nº 334 | Maria das Dores B. Moura Oliveira and Aurora A.C. Teixeira, "*Policy approaches regarding technology transfer: Portugal and Switzerland compared*", September 2009 |
| Nº 333 | Ana Sofia Ferreira, Leonídio Fonseca and Lilian Santos, "*Serão os 'estudantes empreendedores' os empreendedores do futuro? O contributo das empresas juniores para o empreendedorismo*", August 2009 |
| Nº 332 | Raquel Almeida, Marina Silva and Tiago Soares, "*Coesão Territorial - As relações de fronteira entre Portugal e Espanha*", August 2009 |
| Nº 331 | Custódia Bastos, Suzi Ladeira and Sofia Silva, "*Empreendedorismo nas Artes ou Artes do Empreendedorismo? Um estudo empírico do 'Cluster' da Rua Miguel Bombarda*", August 2009 |
| Nº 330 | Filipe A. Ribeiro, Ana N. Veloso and Artur V. Vieira, "*Empreendedorismo Social: Uma análise via associativismo juvenil*", August 2009 |
| Nº 329 | Argentino Pessoa, "*Outsourcing And Public Sector Efficiency: How Effective Is Outsourcing In Dealing With Impure Public Goods?*", July 2009 |
| Nº 328 | Joana Almodovar, Aurora A.C. Teixeira, "*Conceptualizing clusters through the lens of networks: a critical synthesis*", July 2009 |
| Nº 327 | Pedro Mazeda Gil, Fernanda Figueiredo and Óscar Afonso, "*Equilibrium Price Distribution with Directed Technical Change*", July 2009 |
| Nº 326 | Armando Silva, Ana Paula Africano and Óscar Afonso, "*Which Portuguese firms are more innovative? The importance of multinationals and exporters*", June 2009 |
| Nº 325 | Sofia B. S. D. Castro, João Correia-da-Silva and Pascal Mossay, "*The core-periphery model with three regions*", June 2009 |
| Nº 324 | Marta Sofia R. Monteiro, Dalila B. M. M. Fontes and Fernando A. C. C. Fontes, "*Restructuring Facility Networks under Economy of Scales*", June 2009 |
| Nº 323 | Óscar Afonso and Maria Thompson, "*Costly Investment, Complementarities and the Skill Premium*", April 2009 |
| Nº 322 | Aurora A.C. Teixeira and Rosa Portela Forte, "*Unbounding entrepreneurial intents of university students: a multidisciplinary perspective*", April 2009 |
| Nº 321 | Paula Sarmento and António Brandão, "*Next Generation Access Networks: The Effects of Vertical Spillovers on Access and Innovation*", April 2009 |
| Nº 320 | Marco Meireles and Paula Sarmento, "*Incomplete Regulation, Asymmetric Information and Collusion-Proofness*", April 2009 |
| Nº 319 | Aurora A.C. Teixeira and José Sequeira, "*Determinants of the international influence of a R&D organisation: a bibliometric approach*", March 2009 |
| Nº 318 | José Sequeira and Aurora A.C. Teixeira, "*Assessing the influence of R&D institutions by mapping international scientific networks: the case of INESC Porto*", March 2009 |
| Nº 317 | João Loureiro, Manuel M. F. Martins and Ana Paula Ribeiro, "*Cape Verde: The Case for Euroization*", March 2009 |
| Nº 316 | Ester Gomes da Silva and Aurora A.C. Teixeira, "*Does structure influence growth? A panel data econometric assessment of 'relatively less developed' countries, 1979-2003*", March 2009 |

| | |
|---|---|
| Nº 315 | Mário A. P. M. Silva, "*A Model of Growth with Intertemporal Knowledge Externalities, Augmented with Contemporaneous Knowledge Externalities*", March 2009 |
| Nº 314 | Mariana Lopes and Aurora A.C. Teixeira, "*Open Innovation in firms located in an intermediate technology developed country*", March 2009 |
| Nº 313 | Ester Gomes da Silva, "*Capital services estimates in Portuguese industries, 1977-2003*", February 2009 |
| Nº 312 | Jorge M. S. Valente, Maria R. A. Moreira, Alok Singh and Rui A. F. S. Alves, "*Genetic algorithms for single machine scheduling with quadratic earliness and tardiness costs*", February 2009 |
| Nº 311 | Abel Costa Fernandes, "*Explaining Government Spending: a Cointegration Approach*", February 2009 |
| Nº 310 | João Correia-da-Silva, "*Uncertain delivery in markets for lemons*", January 2009 |
| Nº 309 | Ana Paula Ribeiro, "*Interactions between Labor Market Reforms and Monetary Policy under Slowly Changing Habits*", January 2009 |
| Nº 308 | Argentino Pessoa and Mário Rui Silva, "*Environment Based Innovation: Policy Questions*", January 2009 |
| Nº 307 | Inês Drumond and José Jorge, "*Basel II Capital Requirements, Firms' Heterogeneity, and the Business Cycle*", January 2009 |
| Nº 306 | Adelaide Maria Figueiredo, Fernanda Otília Figueiredo and Natália Pimenta Monteiro, "*Labor adjustments in privatized firms: a Statis approach*", December 2008 |
| Nº 305 | Manuela A. D. Aguiar and Sofia B. S. D. Castro, "*Chaotic and deterministic switching in a two-person game*", December 2008 |
| Nº 304 | Ana Pinto Borges and João Correia-da-Silva, "*Using Cost Observation to Regulate Bureaucratic Firms*", December 2008 |
| Nº 303 | Miguel Fonseca, "*The Investment Development Path Hypothesis: a Panel Data Approach to the Portuguese Case*", December 2008 |
| Nº 302 | Alexandre Almeida, Cristina Santos and Mário Rui Silva, "*Bridging Science to Economy: The Role of Science and Technologic Parks in Innovation Strategies in "Follower" Regions*", November 2008 |
| Nº 301 | Alexandre Almeida, António Figueiredo and Mário Rui Silva, "*From Concept to Policy: Building Regional Innovation Systems in Follower Regions*", November 2008 |
| Nº 300 | Pedro Quelhas Brito, "*Conceptualizing and illustrating the digital lifestyle of youth*", October 2008 |
| Nº 299 | Argentino Pessoa, "*Tourism and Regional Competitiveness: the Case of the Portuguese Douro Valley*", October 2008 |
| Nº 298 | Aurora A.C. Teixeira and Todd Davey, "*Attitudes of Higher Education students to new venture creation: a preliminary approach to the Portuguese case*", October 2008 |
| Nº 297 | Carlos Brito, "*Uma Abordagem Relacional ao Valor da Marca*", October 2008 |
| Nº 296 | Pedro Rui M. Gil, Paulo Brito and Óscar Afonso, "*A Model of Quality Ladders with Horizontal Entry*", October 2008 |
| Nº 295 | Maria Manuel Pinho, "*The political economy of public spending composition: evidence from a panel of OECD countries*", October 2008 |
| Nº 294 | Pedro Cosme da Costa Vieira, "*O Subsídio de Desemprego e a Relação Negativa entre Salário e Risco de Falência: Uma Teoria em Equilíbrio Parcial*", October 2008 |
| Nº 293 | Cristina Santos, Alexandre Almeida and Aurora A.C. Teixeira, "*Searching for clusters in tourism. A quantitative methodological proposal*", September 2008 |

*Editor: Sandra Silva (sandras@fep.up.pt)*
*Download* available at:
http://www.fep.up.pt/investigacao/workingpapers/
also in http://ideas.repec.org/PaperSeries.html

www.fep.up.pt

**FACULDADE DE ECONOMIA DA UNIVERSIDADE DO PORTO**
Rua Dr. Roberto Frias, 4200-464 Porto | Tel. 225 571 100
Tel. 225571100 | www.fep.up.pt