

## Hybrid genetic algorithm for the economic lot-scheduling problem

I. MOON<sup>†\*</sup>, E. A. SILVER<sup>‡</sup> and S. CHOI<sup>§</sup>

The economic lot-scheduling problem (ELSP) is an important production scheduling problem that has been intensively studied over 40 years. Numerous heuristic algorithms have been developed since the problem is NP-hard. Dobson's heuristic has been regarded as the best in its performance. The present paper provides a hybrid genetic algorithm based on the time-varying lot sizes approach in the ELSP literature. Numerical experiments show that the hybrid genetic algorithm outperforms Dobson's heuristic.

### 1. Introduction

It is common in industry to produce several products on a single facility (or machine) due to economies of scale. Typically, these facilities can only produce one product at a time, and have to be stopped and prepared (i.e. set-up) at a cost of time and money, before the start of the production run of a different product. A production scheduling problem arises because of the need to coordinate the set-ups and the production runs of the products. The economic lot-scheduling problem (ELSP) is the problem of scheduling production of several products on a single facility, so that level demands are met without stockouts or backorders, and the long run average inventory carrying and set-up costs are minimized. This problem occurs in many production situations including the following (Boctor 1987).

- Metal forming and plastics production lines (press lines, and plastic and metal extrusion machines), where each product requires a different die to be set up on the machine.
- Assembly lines, which produce several products and/or different product models (electric appliances, motor cars, etc.).
- Blending and mixing facilities (for paints, beverages, animal food, etc.), in which different products are poured into different containers.
- Weaving production lines (for textiles, carpets, etc.), in which the main products are manufactured in different colours, widths and grades.

Typically, it is more economical to purchase one high-speed machine capable of producing a number of products than to purchase many dedicated machines. This situation leads to the question of how one should schedule production on this high-speed machine. The issue is one of selecting both a sequence, in which the products

Revision received July 2001.

<sup>†</sup> Department of Industrial Engineering, Pusan National University, Pusan 609-735, Korea.

<sup>‡</sup> Faculty of Management, University of Calgary, Calgary, Canada.

<sup>§</sup> Department of Manufacturing Engineering, University of Calgary, Calgary, Canada.

\* To whom correspondence should be addressed. e-mail: ikmoon@pusan.ac.kr

will be manufactured, and a batch size for each product run. The issue of batching arises because the system usually incurs a set-up cost and/or a set-up time when the machine switches from one product to a different product. The cost may be due to cleaning or to scrap losses occurring when machine settings are adjusted for the next product. Set-up times imply a downtime during which the machine cannot produce, which, in turn, implies a need to carry more inventory. This problem has attracted the attention of many researchers over 40 years, partly because it is a representation of many frequently encountered scheduling problems, and simply because it seems to be very difficult to solve.

In the ELSP, it is typically assumed that production and demand rates are known product-dependent constants, while set-up times and set-up costs are known product-dependent, but sequence-independent constants. In addition, research on the ELSP has focused on cyclic schedules, i.e. schedules that are repeated periodically. Because of its non-linearity, combinatorial characteristics and complexity, the ELSP is generally known as an NP-hard problem (Hsu 1983). Many heuristic approaches have been developed for this problem. Basically, there are three types of approaches.

- (1) Common cycle approach: restricts all the products' cycle times to equal length (an item's cycle time is the duration between the starts of two consecutive runs of that item). Then it finds the optimal common cycle time. This approach has the advantage of always finding a feasible schedule using a very simple procedure. This procedure, however, gives solutions far from the lower bound in some situations.
- (2) Basic period approach: allows different cycle times for different products, but restricts each product's cycle time to be an integer multiple  $k$  of a time period called a basic period. All lots of each item are of the same size. Under this approach, it is NP-hard to find a feasible solution, given the number of production runs per cycle for each of the items. This approach, in general, gives better solutions than the common cycle approach. However, its main drawback is the difficulty of ensuring feasibility.
- (3) Time-varying lot sizes approach: allows different lot sizes for any given product during a cyclic schedule. It explicitly handles the difficulties caused by set-up times and always gives a feasible schedule as proved by Dobson (1987). This approach usually gives better solutions than the previous two approaches.

Khouza *et al.* (1998) successfully applied genetic algorithms to solve the ELSP. Their algorithm is based on the basic period approach since it is ideally suited for using genetic algorithms. Even though the approach itself is meaningful, the results are not encouraging. The deviation from a lower bound was sometimes  $> 80\%$  for the famous Bomberger (1966) data. It could not improve over Dobson's (1987) heuristic, which has had the best performance up to now. (Note that the maximum deviation from a lower bound using Dobson's heuristic is  $< 10\%$ .) The cause of this result is that their algorithm is based on the basic period approach whose performance is inferior to the time-varying lot sizes approach. The purpose of the current research is to develop a hybrid genetic algorithm to solve the ELSP. Our genetic algorithm is based on the time-varying lot sizes approach.

The organization of the paper is as follows. A literature review on the ELSP is presented in Section 2. Section 3 provides assumptions and notation for the ELSP,

and it reviews the modified Dobson's heuristic. In addition, we point out how we may improve upon Dobson's heuristic using a genetic algorithm. Section 4 develops a hybrid genetic algorithm and explains the algorithm using well-known numerical examples. In Section 5, computational tests are performed to compare the performances of the hybrid genetic algorithm with those of Dobson's heuristic. Finally, there are concluding remarks in Section 6.

## 2. Literature review on the ELSP

The ELSP has been extensively studied over 40 years, and more than 100 papers have been published in a variety of journals. We now review a small portion of the literature on the ELSP. The earliest contributions to this problem include Eilon (1957), Rogers (1958) and Hanssmann (1962). A lower bound on the minimum average cost can be easily obtained by taking each product in isolation and calculating economic production quantities, and this approach is known as the independent solution (IS) since it ignores the capacity issue of the sharing of the machine by several products. A tight lower bound has been implicitly suggested by Bomberger (1966), and rediscovered in several different ways by several researchers (Dobson 1987, Gallego and Moon 1992). The idea is to compute economic production quantities under a constraint on the capacity of a machine. The capacity constraint is that enough time must be made available for set-ups. The problem can be formulated as a non-linear program and easily solved via a line search algorithm. However, the synchronization constraint, stating that no two items can be scheduled to produce at the same time, is ignored. Thus, the value of the non-linear program results in a lower bound on the minimum average cost.

Research on the ELSP has focused on cyclic schedules. Moreover, almost all researchers have restricted their attention to cyclic schedules that satisfy the zero switch rule (ZSR). This rule states that a production run for any particular product can be started only if its physical inventory is zero. Counterexamples to the optimality of this rule have been found but are rare (Maxwell 1964, Delporte and Thomas (1978)).

If we restrict ourselves to the case that the cycle times for all products must be the same, the problem is a simpler version of the ELSP and known as the common cycle approach. The objective value obtained from this approach serves as the upper bound on the general ELSP. Jones and Inman (1989) and Gallego (1990) showed that this approach works well under certain situations.

There are two other approaches for heuristics for the ELSP: the basic period approach and the time-varying lot sizes approach. The basic period approach requires, in addition to the ZSR, that every item must be produced at equally spaced intervals that are multiples of a basic period (this together with the ZSR implies that each item is produced in equal lot sizes). Most of the heuristic algorithms that follow this approach first select the frequency (i.e. number of production runs per cycle) with which each product is to be produced, and then search for a feasible schedule that implements these frequencies (Doll and Wybark 1973). See Elmaghraby (1978) for an excellent review on this approach up to the late 1970s. Under this approach, it is NP-complete to determine the existence of a feasible schedule (Hsu 1983). These difficulties have led some researchers to reject the basic period paradigm, in particular the requirement of equally spaced production lots.

The time-varying lot sizes approach, which relaxes the restriction of equally spaced production runs, was initiated by Maxwell (1964) and Delporte and Thomas (1978). Dobson (1987) showed that any production sequence (i.e. the order in which the products are produced in a cycle) can be converted into a feasible production schedule in which the quantities and timing of production lots are not necessarily equal provided that, beyond the time needed for production, there is some time available for setups. Dobson also developed a heuristic to generate production frequencies and a sensible production sequence. Near optimal schedules can be obtained by combining Dobson's heuristic with Zipkin's (1991) algorithm which finds the production run times and machine idle times for each product for a given production sequence. Gallego and Roundy (1992) extended the time-varying lot sizes approach to the ELSP which allows backorders. Dobson (1992) extended his earlier work (1987) allowing the set-up time to be sequence dependent. Gallego and Shaw (1997) showed that the ELSP is strongly NP-hard under the time-varying lot sizes approach with or without the ZSR restriction, giving theoretical justification to the development of heuristics.

As pointed out by Silver (1993) in his review, if quantitative models are to be more useful as aids for managerial decision-making, they must represent more realistic problem formulations, particularly permitting some of the usual *givens* to be treated as decision variables. Givens can be defined as the parameters which have been treated as fixed or given, for example, setup time, setup cost, production rate, defective rate, etc. Silver (1993) listed a wide variety of possible improvements to undertake (equivalently, usual givens to change) in manufacturing operations, such as set-up time/cost reduction, higher quality level, controllable production rates, lead time reduction, etc. There is a rapidly growing literature on modelling the effects of changing the givens in manufacturing decisions. In the realm of changing the givens, a variety of modifications on the ELSP have been developed (Silver *et al.* 1998). Allen (1990) modified the ELSP to allow production rates to be decision variables. He then developed a graphical method for the rates and cycle times for a two-product problem. Silver (1990), Moon *et al.* (1991), Gallego (1993), Khouza (1997), and Moon and Christy (1998) showed that production rate reduction was more profitable for underutilized facilities. Silver (1995) and Viswanathan and Goyal (1997) considered the situation in which a family of products follows a cyclic schedule, but there is a limit on shelf life. The cycle length and production rate are adjusted to ensure a feasible schedule.

Gallego and Moon (1992) examined a multiple product factory that employs a cyclic schedule to minimize holding and set-up costs. When set-up times can be reduced, at the expense of set-up costs, by externalizing internal set-up operations, they showed that dramatic savings are possible for highly utilized facilities. Gallego and Moon (1995) developed an ELSP with the assumptions that set-up times can be reduced by a one time investment. Hwang *et al.* (1993) and Moon (1994) developed an ELSP in which both set-up reduction and quality improvement can be achieved through investment. More recently, Moon *et al.* (1998) applied the stabilization period concept, in which yield rates gradually increase during the period, to the ELSP.

### 3. Hybrid genetic algorithm

#### 3.1. Assumptions and notation

The following assumptions are used in the ELSP.

- (1) Multiple items compete for the use of a single facility.
- (2) Demand rates, production rates, set-up costs and set-up times for all items are known constants.
- (3) Backorders are not allowed.

The following notation is used in the model.

item index  $i = 1, 2, \dots, m$ ,  
 position index  $j = 1, 2, \dots, n$ ,  
 constant production rate (units per day)  $p_i$   $i = 1, 2, \dots, m$ ,  
 constant demand rate (units per day)  $d_i$   $i = 1, 2, \dots, m$ ,  
 inventory holding cost (\$ per unit per day)  $h_i$   $i = 1, 2, \dots, m$ ,  
 set-up cost (\$)  $A_i$   $i = 1, 2, \dots, m$ ,  
 set-up time (days)  $s_i$   $i = 1, 2, \dots, m$ ,  
 item produced at position  $j$   $f^j$   $j = 1, 2, \dots, n$ ,  
 production time duration for item produced at position  $j$   $t^j$   $j = 1, 2, \dots, n$ , and  
 idle time duration after the production of the item at position  $j$   
 $u^j$   $j = 1, 2, \dots, n$ ,  
 cycle length (days)  $T$ .

The ELSP can be stated as follows. There is a single facility on which  $m$  distinct products are to be produced. We try to find a cycle length  $T$ , a production sequence  $\mathbf{f} = (f^1, \dots, f^n)$ , where  $f^j \in \{1, \dots, m\}$ , production time durations  $\mathbf{t} = (t^1, \dots, t^n)$ , and idle time durations  $\mathbf{u} = (u^1, \dots, u^n)$ , so that the production sequence can be completed in the chosen cycle, the cycle can be repeated over time, demand can be fully met, and the total of inventory and set-up costs is minimized (Silver *et al.* 1998). Define

$$\kappa = 1 - \sum_{i=1}^m \frac{d_i}{p_i}.$$

Note that  $\kappa$  is the long-run proportion of time available for set-ups. For infinite horizon problems  $\kappa > 0$  is a necessary condition for the existence of a feasible schedule. Dobson (1987) showed that if  $\kappa > 0$ , then any production sequence can be converted into a feasible schedule by allowing time-varying production runs and a sufficiently large cycle length.

### 3.2. Algorithm

We first represent Dobson's original formulation of the problem. Here subscript  $i$  is used to indicate product  $i$  and superscript  $j$  indicates the product produced at the  $j$ th position in the sequence. Let  $\mathcal{F}$  be the set of all possible finite sequences of products. Here,  $J_i$  denotes the positions in a given sequence where product  $i$  is produced, that is,  $J_i = \{j | f^j = i\}$ . Let  $L_k$  be the positions in a given sequence from  $k$ , up to but not including the position in the sequence where product  $f^k$  is produced again. The complete formulation of the ELSP is

$$\inf_{j \in \mathcal{F}} \min_{t \geq 0, u \geq 0, T \geq 0} \frac{1}{T} \left( \sum_{j=1}^n \frac{1}{2} h^j (p^j - d^j) \left( \frac{p^j}{d^j} \right) (t^j)^2 + \sum_{j=1}^n A^j \right) \quad (1)$$

subject to

$$\sum_{j \in J_i} p_i t^j = d_i T, \quad i = 1, \dots, m \quad (2)$$

$$\sum_{j \in L_k} (t^j + s^j + u^j) = (p^k/d^k)t^k, \quad k = 1, \dots, n \quad (3)$$

$$\sum_{j=1}^n (t^j + s^j + u^j) = T. \quad (4)$$

Constraints (2) ensure that we allocate enough time to each product  $i$  to meet its demand,  $d_i T$ , over the cycle. Constraints (3) mean that we must produce enough of a product each time to last until the next time that that product is produced. Constraint (4) simply states that the cycle time  $T$  must be the sum of production, setup, and idle times for all the items produced in the cycle.

Our hybrid genetic algorithm to solve above problem can be described as follows.

*Step 1.* Find the production frequencies by solving the following LB model. This lower bound is tighter than that obtained by using the so-called independent solution in which each product is taken in isolation by calculating its economic production quantity. The idea instead is to compute economic production quantities under a constraint on the capacity of the machine. The capacity constraint is that enough time must be made available for set-ups. Since the long-run average proportion of time spent on set-ups is  $\sum_i s_i/T_i$ , where  $T_i$  is the cycle length for item  $i$ , and the proportion of time available for set-ups is  $\kappa$ , the capacity constraint is as in (5) below. However, the synchronization constraint, stating that no two items can be scheduled to produce at the same time, is ignored. Consequently, the value of the following nonlinear program results in a lower bound on the total daily cost for the general ELSP. This lower bound scheme has been originally suggested by Bomberger (1966), and rediscovered by several researchers in different ways (Dobson 1987, Gallego and Moon 1992).

### LB

$$\min_{T_1, \dots, T_m} \sum_{i=1}^m \left[ \frac{A_i}{T_i} + \frac{h_i d_i T_i}{2} \left( 1 - \frac{d_i}{p_i} \right) \right]$$

subject to

$$\sum_{i=1}^m \frac{s_i}{T_i} \leq \kappa \quad (5)$$

$$T_i \geq 0 \quad i = 1, \dots, m.$$

The objective function and the constraint set, in the above model, are convex in the  $T_i$ 's. Therefore, the optimal points of the LB model are points which satisfy the Karush–Kuhn–Tucker (KKT) conditions as follows:

$$T_i = \sqrt{\frac{A_i + \lambda s_i}{H_i}} \quad \forall i$$

$$\lambda \geq 0 \quad \text{complementary slackness with } \sum_{i=1}^m \frac{s_i}{T_i} \leq \kappa,$$

where  $H_i = h_i d_i (1 - (d_i/p_i))/2$ . We can use the following procedure to find the optimal  $T_i$ .

### Algorithm for lower bound

- (Step 1) Check if  $\lambda = 0$  gives an optimal solution.  
Find  $T_i$ s from following equations  $T_i = \sqrt{A_i/H_i} \forall i$ .
- (Step 2) If  $\sum_{i=1}^m (s_i/T_i) \leq \kappa$ , then the  $T_i$ s are an optimal solution.  
Otherwise, go to Step 3.
- (Step 3) Start with an arbitrary  $\lambda > 0$ .
- (Step 4) Compute  $T_i = \sqrt{(A_i + \lambda s_i)/H_i} \forall i$ .
- (Step 5) If  $\sum_{i=1}^m (s_i/T_i) < \kappa$ , reduce  $\lambda$ . Go to Step 4.  
If  $\sum_{i=1}^m (s_i/T_i) > \kappa$ , increase  $\lambda$ . Go to Step 4.  
If  $\sum_{i=1}^m (s_i/T_i) = \kappa$ , stop. The  $T_i$ s are optimal.

Let the optimal cycle length for item  $i$  in program LB be  $T_i^*$ . Also let  $x_i$  represent the relative production frequency for item  $i$ . Then  $x_i$  is determined by the following:

$$x_i = \frac{\max_j \{T_j^*\}}{T_j^*} \quad i = 1, 2, \dots, m.$$

- Step 2. Round the production frequencies obtained in Step 1 to the nearest integers.
- Step 3. We use a genetic scheme to find a good production sequence using the production frequencies obtained in Step 2. We will explain the details of the genetic scheme in the next section.
- Step 4. Solve for  $\mathbf{t}$  and  $\mathbf{u}$ , given  $\mathbf{f}$ . If we assume that there are no idle times (that is,  $\mathbf{u} = 0$ ) for a given production sequence  $\mathbf{f}$ , we can find  $\mathbf{t}$  using equation (3). This approximation works very well for a highly loaded facility. This method is called a quick-and-dirty heuristic, and we use this method in the study. Otherwise, we can use the parametric algorithm as in Zipkin (1991).

Note that our algorithm is different from Dobson's heuristic in Steps 2 and 3. The following steps are Dobson's Steps 2 and 3.

- Step 2. Round off the frequencies to power-of-two integers. It has been shown by Roundy (1989) that additional costs do not exceed 6% when we convert the real values of production frequencies to power-of-two integers. The conversion of production frequencies to power-of-two integers enables the determination of the production sequences to be easily accomplished in Step 3. Let  $y_i$  be the production frequency for item  $i$ , which is a power-of-two integer.
- Step 3. Find a production sequence using the bin-packing heuristic suggested by Doll and Whybark (1973) and Dobson (1987). Given these new frequencies  $\mathbf{y}$ , the bin-packing heuristic attempts to spread them out as evenly as possible. In particular, create  $b$  bins in which  $b = \max_i \{y_i\}$ . For each product  $i$ ,

estimate the production time duration  $v_i$  for the lots by assuming that the lots will be equally spaced. That is, we compute  $v_i = s_i + (d_{it}/p_i y_i)$ . Now we regard the problem as that of bin-packing with  $b$  bins and  $y_i$  items of height  $v_i$  for all  $i$ . The only additional restriction is that we put items in bins equally spaced. If  $b = 4$  and  $y_i = 2$ , we have two choices: either bins 1, 3 or 2, 4. When we assign items to the bins, we use a variation of the longest processing time (LPT) rule in which items are ordered lexicographically by  $(y_i, v_i)$ . The sequences given for the bins were strung together to provide a production sequence which we need in this step. By minimizing the maximum height of the bin, the heuristic finds an efficient production sequence  $\mathbf{f}$  (see Dobson 1987 for more details).

Now we explain how our hybrid genetic algorithm can improve upon Dobson's heuristic. First, why is the rounding-off procedure required in the heuristic? The reason is that the production frequencies must be power-of-two integers in order to apply the classical bin-packing heuristic to find a production sequence. Even though we know that the penalty of rounding does not cause more than 6% additional cost, it may be critical when we want to save 1% in total cost.

Thus, our genetic algorithm uses production frequencies rounded to the nearest integers rather than those rounded to the power-of-two integers. In addition, we use a genetic scheme to find a good production sequence rather than using the bin-packing heuristic. However, we use the same lower bound scheme as in Dobson's heuristic to find the  $x_i$ s. That's why we call this algorithm a *hybrid* genetic algorithm.

#### 4. Finding a production sequence using the genetic algorithm

This section explains the Step 3 of the hybrid genetic algorithm more detail. The main ideas of a genetic algorithm are introduced shortly and how we adapt a genetic algorithm to our problem will be shown.

Genetic algorithms, which have been widely used in various areas for three decades, are stochastic search algorithms based on the mechanism of natural selection and natural genetics. Genetic algorithms, differing from conventional search techniques, start with an initial set of (random) solutions called a population. Each individual in the population is called a chromosome, representing a solution to the problem at hand. The chromosomes evolve through successive iterations, called generations. During each generation, the chromosomes are evaluated, using some measures of fitness. Generally speaking, the genetic algorithm is applied to spaces which are too large to be exhaustively searched.

It is generally accepted that any genetic algorithm to solve a problem must have basic components, but have different characteristics depending on the problem under study. We explain our overall strategies including chromosome style as follows.

- Representation and initialization.
- Objective and fitness function.
- Reproduction, crossover and mutation.
- Fitness scaling.



#### 4.1. Representation and initialization

The proper representation of a solution plays a key role in the development of a genetic algorithm. A string consists of real integers is a solution (a chromosome) in this paper. Chromosome length is the sum of the production frequencies (each rounded to the nearest integers) and each gene in the chromosome is the index of a product. Since creating the chromosome satisfying the given frequencies is not easy, we introduce an extra chromosome, which provides the absolute location of each of the genes in the first chromosome. Thus, we use two kinds of chromosomes. Info A represents the product numbers, and Info B indicates the absolute locations of the genes. For example, suppose we have a five-item problem with the items having to be produced 2, 2, 3, 3, and 1 times during a cycle, respectively. Then, we have the following chromosome information and its length is 11 (figure 1).

Each gene in Info A has to be an integer between 1 and 5, and the number in Info B identifies the relative position in the chromosome. We should refer to Info A when we convert the chromosome to a production sequence. That is, a chromosome (5, 3, 8, 6, 1, 9, 4, 7, 11, 10, 2) means actually the chromosome (3, 2, 4, 3, 1, 4, 2, 3, 5, 4, 1).

#### 4.2. Objective and fitness function

GALib, which is very popular GA program and will be used in this study, provides various kinds of objective functions. Thus, the only thing we have to do is to select an appropriate objective function. In this study, we choose minimization function, `ga.minimize()`. A fitness function is computed for each string (i.e. chromosome) in the population and the objective is to find a string with the minimum fitness function value. For any given string, we can find the production time durations from Step 4 of the main algorithm of Section 3.2 and compute the fitness function of the chromosome from the objective function (equation 1).

#### 4.3. Reproduction, crossover and mutation

A simple genetic algorithm that yields good results in many practical problems is composed of three operators: reproduction, crossover and mutation. Reproduction is a process in which individual strings are copied according to their objective functions.

The reproduction operator may be implemented in algorithmic form in a number of ways. Perhaps the easiest is to create a biased roulette wheel where each current string in the population has a roulette wheel slot sized in proportion to its fitness. Stochastic tournament method will be adopted in this paper. In the method, selection probabilities are calculated as above and successive pairs of individuals are drawn using roulette wheel selection. A pair is drawn and the string with the higher fitness is declared the winner. The string is inserted in the new population. Another pair is drawn, etc. This process continues until the population is full.

The crossover operator takes two chromosomes and swaps a part of their genetic information to produce new chromosomes. A simple way to achieve crossover would

Info A :	1	1	2	2	3	3	3	4	4	4	5
Info B :	1	2	3	4	5	6	7	8	9	10	11

Figure 1. Structure of a chromosome.

be to choose a random cut-point and generate the offspring by combining the segment of one parent to the left of the cut-point with the segment of the other parent to the right of the cut-point.

In this study, however, we deal with a sequence chromosome, not the binary one. We need a more complicated crossover operator which enables us to prevent two or more genes from having the same value in Info B. Various useful crossover operators for this purpose have been developed, and we use PMX (partial matched crossover). PMX has been developed to tackle a traveling salesman problem (Goldberg 1989). Under PMX, two strings are aligned, and two crossing points are picked uniformly at random along the strings. These two points define a matching selection that is used to effect a cross through position-by-position exchange operations. To see this, consider the two strings in figure 2.

PMX proceeds by positionwise exchanges. First, mapping string P2 to string P1, 1 and 2, 9 and 7, 6 and 5, and 4 and 3 exchange places. Similarly mapping string P1 to string P2, the other pairs of 2 and 1, 7 and 9, 5 and 6, and 3 and 4 exchange places (as indicated in **bold**). Through the mapping procedure, O1 and O2 have been developed.

Mutation is a background operator, which produces spontaneous random changes in various chromosomes. A simple way to achieve mutation is to alter one or more genes. In genetic algorithms, mutation serves the crucial role of replacing the genes lost from the population during the selection process so that they can be tried in a new context, or providing the genes that were not present in the initial population. In a sequenced string, mutation should occur in a pair of genes since each number represents a position in a sequence. For example, if a number 9 has been mutated to a number 7, the number 7 in the original chromosome must be changed to a number 9 in the new chromosome. This is illustrated in figure 3.

#### 4.4. *Fitness scaling*

Since the early stages of genetic algorithm study, scaling of objective function values has been widely accepted in practice. This is done to keep appropriate levels of competition throughout a search. Without scaling, early on there is a tendency for a few superindividuals to dominate the selection process. In our problem, the objective functions must be scaled back to prevent domination of the population by these superstrings. Later on, when the population is largely converged, competition among

P1	8	9	4		2	7	5	3		11	10	1	6
P2	2	8	10		1	9	6	4		3	7	5	11
⇓													
O1	8	<b>7</b>	<b>3</b>		1	9	6	4		11	10	<b>2</b>	<b>5</b>
O2	<b>1</b>	8	10		2	7	5	3		<b>4</b>	<b>9</b>	<b>6</b>	11

Figure 2. PMX (partial matched crossover).

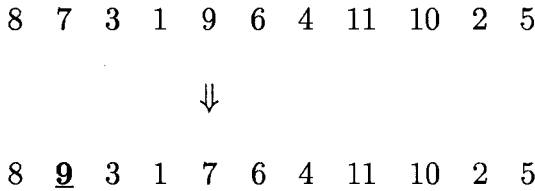


Figure 3. Mutation.

population members is less strong and the search tends to wander. In this case, the objective functions must be scaled up to accentuate differences between population members to continue to reward the best performers. There are various scaling methods (linear scaling, sigma ( $\sigma$ ) truncation, power law scaling, logarithmic scaling, etc.) (Gen and Cheng 1997). We apply sigma truncation developed by Forrest (1985) to the ELSP. It uses population variation information (the standard deviation,  $\sigma$ ) to preprocess raw fitness values prior to scaling.

### 5. Computational experiments

**Example 1:** First we solve Mallya’s five-item problem (1992) to compare the hybrid genetic algorithm with the modified Dobson’s heuristic. The data for Mallya’s problem are shown in table 1 ( $h_i = ic_i$ ).

We first show the detailed steps of the modified Dobson’s heuristic as follows:

(Step 1) We compute a lower bound and associated cycle length for each item.

$$T_1 = 45.06, T_2 = 73.56, T_3 = 33.53, T_4 = 41.79, T_5 = 112.41$$

with a lower bound \$57.73.

(Step 2) Compute production frequencies rounded to power-of-two integers:

$$y_1 = 2, y_2 = 2, y_3 = 4, y_4 = 2, y_5 = 1.$$

(Step 3) Obtain a production sequence using the bin-packing heuristic. We use the following lexicographic order to assign items to the bins:

$$\begin{aligned}
 (y_3, v_3) = (4, 3.845) &\geq^L (y_4, v_4) = (2, 17.483) \geq^L (y_1, v_1) = (2, 14.943) \\
 &\geq^L (y_2, v_2) = (2, 9.599) \geq^L (y_5, v_5) = (1, 12.542).
 \end{aligned}$$

Product	Production rate ( $p_i$ ) (units/day)	Demand rate ( $d_i$ ) (units/day)	Set-up time ( $s_i$ ) (days)	Set-up cost ( $A_i$ ) (money unit)	Standard cost ( $c_i$ ) (money unit)
1	1800	474	0.20	80	0.00379
2	2500	413	0.35	140	0.00252
3	4000	528	0.15	60	0.00391
4	3200	985	0.25	100	0.00282
5	1500	166	0.15	60	0.00108

Table 1. Data for Mallya’s example.

Four lots for product 3 are the first assigned to each of four bins. Next two lots for product 4 are assigned to bins 1 and 3. Then, two lots for product 1 are assigned to bins 2 and 4. Two lots for product 2 are assigned again to bins 2 and 4 since we want to minimize the maximum height of the bin. Finally the lot for product 5 is assigned to bin 1. The resulting production sequence is as follows:

$$\mathbf{f} = (3, 4, 5, 3, 1, 2, 3, 4, 3, 1, 2).$$

(Step 4) Compute production time durations and total average daily cost:

$$\mathbf{t} = (4.655, 17.666, 12.392, 3.190, 11.880, 8.399, 2.616, 16.800, 4.320, 17.606, 10.099)$$

with total average daily cost \$61.63.

We now show the detailed steps of the hybrid genetic algorithm as follows:

(Step 1) We compute a lower bound and associated cycle lengths for each item:

$$T_1 = 45.06, T_2 = 73.56, T_3 = 33.53, T_4 = 41.79, T_5 = 112.4$$

with a lower bound \$57.73.

(Step 2) Compute production frequencies rounded to the nearest integers:

$$y_1 = 2, y_2 = 2, y_3 = 3, v_4 = 3, y_5 = 1.$$

(Step 3) Obtain a production sequence using the genetic algorithm:

$$\mathbf{f} = (3, 2, 4, 3, 1, 4, 2, 3, 5, 4, 1).$$

(Step 4) Compute production time durations and total average daily cost:

$$\mathbf{t} = (3.412, 10.093, 11.596, 6.382, 19.094, 12.730, 9.192, 5.615, 12.919, 11.607, 11.647)$$

with total average daily cost \$60.91.

We improved over the modified Dobson's heuristic by 1.1%. The 1% saving is quite meaningful since the average cost penalty of the modified Dobson's heuristic compared with the lower bound is only a few per cent.

**Example 2:** Bomberger's (1966) 10-item problem is quite famous in the ELSP literature. It has been frequently used to compare heuristics. We consider the case of  $\kappa = 0.01$ , which represents a highly loaded facility. Khouza *et al.* (1998) reported that their GA algorithm resulted in \$55 544.57/year, which is equivalent to \$231.44/day since 1 year is assumed to be 240 production days in Bomberger's problem. Our GA algorithm resulted in \$126.12/day (i.e. an 1.8% improvement over Dobson's approach; approximately half the gap to the lower bound) as shown in table 2. The detailed  $\mathbf{f}$  and  $\mathbf{t}$  for both algorithms are shown below.

Lower bound	Present GA	Dobson's heuristic	Khouza's GA	Common cycle solution
122.96	126.12	128.43	231.44	196.14

Table 2. Results for Bomberger's 10-item problem ( $\kappa = 0.01$  case).

(1) Modified Dobson's heuristic:

$\mathbf{f} = (8, 4, 5, 8, 9, 8, 4, 10, 8, 3, 2, 8, 4, 5, 8, 9, 8, 4, 6, 1, 8, 3, 2, 8, 4, 5, 8, 9, 8, 4, 10, 8, 3, 2, 8, 4, 5, 8, 9, 8, 4, 6, 7, 8, 3, 2)$ .

$\mathbf{t} = (30.943, 53.917, 19.470, 34.956, 82.924, 31.630, 49.316, 26.100, 28.670, 42.891, 25.273, 32.171, 56.457, 19.892, 35.461, 84.140, 37.541, 50.513, 12.896, 25.991, 26.511, 39.432, 23.532, 30.298, 52.804, 19.030, 34.216, 81.144, 31.051, 48.139, 25.882, 27.840, 41.560, 24.604, 31.481, 55.101, 19.582, 35.062, 83.180, 34.917, 49.617, 13.095, 19.493, 27.078, 40.272, 24.058)$ .

(2) GA algorithm:

$\mathbf{f} = (8, 9, 5, 8, 4, 2, 3, 8, 10, 4, 8, 5, 9, 8, 2, 4, 8, 3, 6, 1, 5, 8, 9, 4, 2, 8, 3, 4, 5, 8, 9, 8, 10, 4, 8, 2, 5, 3, 8, 7, 6, 4)$ .

$\mathbf{t} = (39.540, 79.095, 14.368, 42.991, 40.219, 19.441, 42.863, 35.904, 28.488, 57.219, 35.913, 14.205, 70.524, 38.638, 20.069, 72.223, 33.637, 28.793, 11.755, 23.560, 15.266, 51.676, 61.846, 35.684, 25.407, 47.645, 39.356, 60.418, 13.582, 37.449, 88.929, 30.922, 18.633, 55.077, 31.503, 23.434, 13.261, 37.791, 36.326, 17.670, 11.805, 56.124)$ .

In addition, we performed computational experiments to compare the performance of our GA with that of Dobson's heuristic. The data set was generated randomly from uniform distributions on the given intervals (table 3), and 50 problems were generated. As it is well known, the ELSP is more meaningful and difficult to solve when  $\kappa$  is small. Thus, we only used problems with  $\kappa \leq 0.1$ . Two kinds of parameters are involved in the computation, i.e., problem data and GA parameters. The former (table 3) include number of items, production rates, demand rates, inventory holding costs, etc. The second set of parameters include the population size, replacement policy, number of generations, crossover rate, and mutation rate. In this study, we performed computational experiments with the following GA parameters.

- Population size: 100
- Elitist strategy (the best individual was always kept from generation to generation).

Parameters	Set
Number of items (units)	[5, 15]
Production rate (units/unit time)	[2000, 20000]
Demand rate (units/unit time)	[1500, 2000]
Set-up times (unit times)	[1, 4]
Set-up cost (\$)	[50, 100]
Holding cost (\$/unit time)	[1/240, 6/240]

Table 3. Distributions for randomly generated data for test problems.

	Mean ratio	Minimum ratio	Maximum ratio
Dobson's heuristic	1.0424	1.0112	1.0889
GA algorithm	1.0302	1.0122	1.0564

Table 4. Computational results for test problems (I): ratio of the solution value to the value of the lower bound.

Mean ratio	Minimum ratio	Maximum ratio
1.0119	0.9835	1.0567

Table 5. Computational results for test problems (II): ratio of the value of Dobson's heuristic to the value of the GA solution.

- Termination condition was to stop the algorithm when the number of generations reaches 1000 (actually most runs of the GA converged before 300 generations) or the best individual does not improve over 150 consecutive generations.
- Crossover rate: 0.9.
- Mutation rate:  $1/(\text{string length of chromosome})$ .

We compare the ratio of the objective value of the modified Dobson's heuristic to a lower bound with the ratio of the objective value of the GA algorithm to the same lower bound (table 4). GA outperformed Dobson's heuristic in 38 out of 50 problems. Table 5 reports the ratio of the objective value of the modified Dobson's heuristic to that of the GA algorithm. It also confirms that GA outperformed Dobson's heuristic. In practice, it is a good idea to use the better of two results after solving a problem with both algorithms.

## 6. Concluding remarks

The economic lot-scheduling problem has been studied by many researchers. It captures many important features of real and frequently encountered scheduling problems. Because of the non-linearity and combinatorial properties of the problem, most researchers have focused on the development of a heuristic algorithm to find a near-optimal solution, which is commonly compared against a lower bound. We have developed a hybrid genetic algorithm that improves over the best heuristic to date in the ELSP literature.

## Acknowledgements

The authors are grateful for the anonymous referees. Research was supported by the Brain Korea (BK) 21 project sponsored by the Ministry of Education in Korea, by the Natural Sciences and Engineering Research Council of Canada (Grant A1485), and by the Carma Chair at the University of Calgary. The paper was awarded a Gold prize for the best paper competition in the 1st Cyber conference hosted by Korean Institute of Industrial Engineers.

## References

- ALLEN, S. J., 1990, Production rate planning for two products sharing a single process facility: a real world case study. *Production and Inventory Management*, **31**, 24–29.
- BOCTOR, F., 1987, The G-group heuristic for single machine lot scheduling. *International Journal of Production Research*, **25**, 363–379.
- BOMBERGER, E., 1966, A dynamic programming approach to a lot size scheduling problem. *Management Science*, **12**, 778–784.
- DELPORTE, C. and THOMAS, L., 1978, Lot sizing and sequencing for N products on one facility. *Management Science*, **23**, 1070–1079.
- DOBSON, G., 1987, The economic lot scheduling problem: achieving feasibility using time-varying lot sizes. *Operations Research*, **35**, 764–771.
- DOBSON, G., 1992, The cyclic lot scheduling problem with sequence-dependent setups. *Operations Research*, **40**, 736–749.
- DOLL, L. and WHYBARK, C., 1973, An iterative procedure for the single machine, multiproduct lot scheduling problem. *Management Science*, **20**, 50–55.
- EILON, S., 1957, Scheduling for batch production. *Institute of Production Engineering Journal*, **36**, 549–579.
- ELMAGHRABY, S., 1978, The economic lot scheduling problem (ELSP): review and extensions. *Management Science*, **24**, 587–598.
- FORREST, S., 1985, *Documentation for PRISONERS DILEMMA and NORMS programs that Use the Genetic Algorithm* (Ann Arbor: University of Michigan).
- GAlib: A C++ Library of Genetic Algorithm Components*, v.2.4.3 Documentation on the Internet [<http://lancet.mit.edu/ga/>].
- GALLEGO, G., 1990, An extension to the class of easy economic lot scheduling problem easy? *IIE Transactions*, **22**, 189–190.
- GALLEGO, G., 1993, Reduced production rates in the economic lot scheduling problem. *International Journal of Production Research*, **31**, 1035–1046.
- GALLEGO, G. and MOON, I., 1992, The effect of externalizing setups in the economic lot scheduling problem. *Operations Research*, **40**, 614–619.
- GALLEGO, G. and MOON, I., 1995, Strategic investment to reduce setup times in the economic lot scheduling problem. *Naval Research Logistics*, **42**, 773–790.
- GALLEGO, G. and ROUNDY, R., 1992, The economic lot scheduling problem with finite back-order costs. *Naval Research Logistics*, **39**, 729–739.
- GALLEGO, G. and SHAW, X., 1997, Complexity of the ELSP with general cyclic schedules. *IIE Transactions*, **29**, 109–113.
- GEN, M. and CHENG, R., 1997, *Genetic Algorithms and Engineering Design* (New York: Wiley).
- GOLDBERG, D. E., 1989, *Genetic Algorithms in Search, Optimization and Machine Learning* (New York: Addison-Wesley).
- HANSSMANN, F., 1962, *Operations Research in Production and Inventory Control* (New York: Wiley).
- HSU, W., 1983, On the general feasibility test for scheduling lot sizes for several products on one machine. *Management Science*, **29**, 93–105.
- HWANG, H., KIM, D. and KIM, Y., 1993, Multiproduct economic lot size models with investment costs for setup reduction and quality improvement. *International Journal of Production Research*, **31**, 691–703.
- JONES, P. and INMAN, R., 1989, When is the economic lot scheduling problem easy? *IIE Transactions*, **21**, 11–20.
- KHOUSA, M., 1997, The economic lot scheduling problem under volume flexibility. *International Journal of Production Economics*, **48**, 73–86.
- KHOUSA, 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**, 509–524.
- MALLYA, R., 1992, Multi-product scheduling on a single machine: a case study. *Omega*, **20**, 529–534.
- MAXWELL, W., 1964, The scheduling of economic lot sizes. *Naval Research Logistics Quarterly*, **11**, 89–124.

- MOON, D. and CHRISTY, D., 1998, Determination of optimal production rates on a single facility with dependent mold lifespan. *International Journal of Production Economics*, **54**, 29–40.
- 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**, 2795–2801.
- MOON, I., GALLEGO, G. and SIMCHI-LEVI, D., 1991, Controllable production rates in a family production context. *International Journal of Production Research*, **29**, 2459–2470.
- MOON, I., HAHM, J. and LEE, C., 1998, The effect of the stabilization period on the economic lot scheduling problem. *IIE Transactions*, **30**, 1009–1017.
- ROGERS, J., 1958, A computational approach to the economic lot scheduling problem. *Management Science*, **4**, 264–291.
- ROUNDY, R., 1989, Rounding off to powers of two in continuous relaxation of capacitated lot sizing problems. *Management Science*, **35**, 1433–1442.
- SILVER, E., 1990, Deliberately slowing down output in a family production context. *International Journal of Production Research*, **28**, 17–27.
- SILVER, E., 1993, Modelling in support of continuous improvements towards achieving world class operations. In *Perspectives in Operations Management* (Massachusetts: Kluwer), pp. 23–44.
- SILVER, E., 1995, Dealing with a shelf life constraint in cyclic scheduling by adjusting both cycle time and production rate. *International Journal of Production Research*, **33**, 623–629.
- SILVER, E., PYKE, D. and PETERSON, R., 1998, *Inventory Management and Production Planning and Scheduling*, 3rd edn (New York: Wiley).
- VISWANATHAN, S. and GOYAL, S. K., 1997, Optimal cycle time and production rate in a family production context with shelf life considerations. *International Journal of Production Research*, **35**, 1703–1711.
- ZIPKIN, P., 1991, Computing optimal lot sizes in the economic lot scheduling problem. *Operations Research*, **39**, 56–63.