International Journal of Production Research,
Vol. 44, No. 21, 1 November 2006, 4551–4568

Hybrid genetic algorithm for group technology economic lot scheduling problem

I. K. MOON*†, B. C. CHA‡ and H. C. BAE†

†Department of Industrial Engineering, Pusan National University, Busan 609-735, Korea
‡Postal Technology Research Center, ETRI, Daejeon 305-700, Korea

(*Revision received December 2005*)

The concept of group technology has been successfully applied to many production systems, including flexible manufacturing systems. In this paper we apply group technology principles to the economic lot scheduling problem, which has been studied for over 40 years. We develop a heuristic algorithm and a hybrid genetic algorithm for the group technology economic lot scheduling problem. Numerical experiments show that the developed algorithms outperform the existing heuristics.

Keywords: Group technology; Economic lot scheduling problem; Genetic algorithm

1. Introduction

In this paper, we consider the problem of scheduling several products on a single facility. A production facility produces several products in order to make use of its utilization. Applying the economic production quantity model on a single facility often produces an infeasible solution due to the constraint of producing no more than one item at any given time. The issue is one of selecting both a sequence in which the products will be manufactured, and a lot or batch size for each product run. The Economic Lot Scheduling Problem (ELSP) is defined as the problem of finding the production sequence, production times and idle times of several products in a single facility on a repetitive basis so that the demands are made without stock-outs or backorders and average inventory holding and setup costs are minimized.

Traditional inventory theory clearly shows that larger setup costs result in larger economic lot sizes. In order to overcome this paradoxical behaviour, we consider group technology (GT) principles. GT is an approach to manufacturing and engineering management that seeks to achieve the efficiency of high-speed and mass production by identifying similar parts and classifying them into groups based on their similarities. The GT approach often has many benefits in manufacturing systems. Several major benefits are shortened setup times, reduced

*Corresponding author. Email: ikmoon@pusan.ac.kr

International Journal of Production Research
ISSN 0020–7543 print/ISSN 1366–588X online
http://www.tandf.co.uk/journals
DOI: 10.1080/00207540500534405

work-in-process inventory, less material handling, and better production planning and control (Burbidge 1988).

Under the GT environment with these benefits, we consider a classical ELSP. The Group Technology Economic Lot Scheduling Problem (GT-ELSP) assumes that a single facility is dedicated to the production of several families of products with the restriction that it can be used to produce one product at a time. The production rates, demand rates, setup cost and times of each item are known to be product-dependent constants. The objective is to determine a feasible production schedule which is repeatable over an infinite planning horizon so that the long-run average setup and inventory holding costs are minimized without stockouts or backorders.

Because of its nonlinearity, combinatorial characteristics, and complexity, the ELSP is generally known as a NP-hard problem (Hsu 1983, Gallego and Shaw 1997). Research on the ELSP has focused on cyclic schedules that are repeated periodically. To solve the ELSP, researchers have followed one of the following two approaches (Elmaghraby 1978).

(1) *Analytic approaches* that achieve the optimum of a restricted version of the original problem. For example, the common cycle approach (Hanssmann 1962) restricts all the products' cycle times to equal length so that this approach always provides a feasible schedule of the restricted version.
(2) *Heuristic approaches* that achieve good solutions of the original problem. These approaches have been found to be more effective in determining the optimal solution than analytic approaches. However, it is not easy to show the feasibility and efficiency of the approaches. Two heuristic approaches exist in the literature, the basic period approach (Bomberger 1966) and the time-varying lot sizes approach (Delporte and Thomas 1977, Dobson 1987, Moon *et al.* 1998). The basic period approach requires that every item must be produced at integer multiple intervals of a basic time period. Under this approach, it is NP-hard to find a feasible schedule. The time-varying lot sizes approach allows different lot sizes for any given product during a cyclic schedule. It explicitly handles the difficulties caused by setup times and always gives a feasible schedule (Dobson 1987).

Heuristic approaches of the ELSP are ideally suited for using genetic algorithms. Khouja *et al.* (1998) applied genetic algorithms for solving the ELSP which is formulated using the basic period approach. Moon *et al.* (2002) developed a hybrid genetic algorithm based on time-varying lot sizes to solve the ELSP, and showed that it outperforms Khouja's genetic algorithm. The purpose of this research is to develop a heuristic and a hybrid genetic algorithm to solve the GT-ELSP based on the group common cycle and time-varying lot sizes approach.

This paper is organized as follows. A literature review of the GT-ELSP is presented in section 2. In section 3, assumptions and notation are presented and we develop a heuristic algorithm. A hybrid genetic algorithm for the ELSP is developed in section 4. In section 5, a numerical example is given to illustrate the solution procedure and to compare the results with other heuristic solutions. We also perform a computational study to compare the performances of the heuristic with those of the genetic algorithm. The paper ends with concluding remarks given in section 6.

2. Literature review of the GT-ELSP

Over the past 40 years, the ELSP and many of its variants have been studied by a large number of researchers. A comprehensive review of the ELSP until the late seventies can be found in Elmaghraby (1978). Recent developments concerning the ELSP are cited in Silver *et al*. (1998). Hsu (1983) showed that the problem is NP-hard as it requires satisfying both the production capacity constraint, stating that enough time must be made available for setups, and the synchronization constraint, stating that only one product can be produced at a time, simultaneously. A tight lower bound has implicitly been 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 problem can be formulated as a nonlinear program and easily solved via a line search algorithm. However, the synchronization constraint is ignored. Thus, the value of the nonlinear program results in a lower bound on the minimum average cost. Kim and Lee (1989) presented production-inventory systems considering probabilistic demand and limited production capacity. Research on the ELSP has focused on 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.

If the cycle times for all products are restricted to be equal, then the approach is known as the common cycle approach (Hanssmann 1962). The guarantee of feasibility is the main advantage of this approach by imposing some constraints on the problem. Jones and Inman (1989) and Gallego (1990) showed that this approach provides optimal or near-optimal solutions under certain conditions.

There are two other heuristic approaches 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 multiple intervals of a basic period (this, together with the ZSR, implies that each item is produced in equal lot sizes). The time-varying lot sizes approach, which relaxes the restriction of equally spaced production runs, was initiated by Maxwell (1964) and Delporte and Thomas (1977). Dobson (1987) developed a formulation of the problem that allows lot sizes and cycle times to vary over time by explicitly taking into account setup times. Zipkin (1991) developed an algorithm to find the production run times and machine idle times for each product taking the production sequence of Dobson (1987) as given.

Studies of GT were originated by Opitz (1970). Several researchers, such as Ham *et al*. (1985) and Wemmerlöv and Hyer (1989), reported some advantages of GT in improving productivity and competitiveness. A heuristic algorithm for classifying similar parts into groups based on their similarities in GT can be found in Ree (1996). The earliest contributions to the GT-ELSP were made by Ham *et al*. (1985). They showed that lot scheduling problems of producing several families on a facility can be treated in an easy way using a common cycle approach to all the items and the groups. Kuo and Inman (1990) developed a practical heuristic to generate a simple and feasible schedule for the GT-ELSP. They also applied the common cycle approach to the groups. Moreover, they allowed different lot sizes for the items within the groups.

The purpose of this research is twofold: (i) to improve the heuristic of Kuo and Inman (1990) by taking into account the modified cycle length in order to balance between the setup and inventory holding costs before and after finding a feasible production duration and (ii) to develop a hybrid genetic algorithm for the GT-ELSP.

3. Heuristic algorithm

3.1 *Heuristic assumptions and notation*

The following assumptions are used in the GT-ELSP.

(1) Items to be processed are classified into several groups based on the group technology concept.
(2) Multiple items within several family groups compete for the use of a single facility.
(3) Major setup costs and times are known to constants.
(4) Demand rates, production rates, setup costs, setup times and holding costs for all items are known to product-dependent constants.
(5) No backlogging of demand is permitted.
(6) Production capacity is sufficient to meet total demand.
(7) The time horizon is infinite.

The following notation is used in the model.

i item index, $i = 1, 2, \ldots, n_j$
j group index, $j = 1, 2, \ldots, J$
k position index of all items in all groups, $k = 1, 2, \ldots, l$
p_{ij} constant production rate of item i of group j (units per day)
d_{ij} constant demand rate of item i of group j (units per day)
h_{ij} known inventory holding cost of item i of group j ($ per unit per day)
A_{ij} known minor setup cost for item i of group j ($)
A_j known major setup cost for group j ($)
s_{ij} known minor setup time for item i of group j (days)
s_j known major setup time for group j (days)
m_{ij} production frequency of item i of group j in each cycle
T cycle length (days)
f^k item produced at position k
t^k production time duration for item produced at position k
u^k machine idle time duration after the production of item at position k

The GT-ELSP can be stated as follows. A major setup of the machine occurs when a group is switching to another and a minor setup occurs when an item is switching to the other item in the same group. The minor setup time and cost are assumed to be small compared with those of the major setup. The problem is to find the production frequency, a cycle length T, a production sequence $\mathbf{f} = (f^1, \ldots, f^l)$ where $f^k \in \{11, \ldots, n_jJ\}$, a production time duration $\mathbf{t} = (t^1, \ldots, t^l)$, and an idle time duration $\mathbf{u} = (u^1, \ldots, u^l)$, of several items that are divided into j groups in a single machine to minimize average setup and inventory holding costs.

Define

$$\gamma = 1 - \sum_{j=1}^{J} \sum_{i=1}^{n_j} \frac{d_{ij}}{p_{ij}}.$$

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

3.2 *Heuristic algorithm*

Like Kuo and Inman (1990), we apply the group common cycle approach and the time-varying lot sizes approach to the items in a group. In the group common cycle approach, the cycle times of all the groups are assumed to be equal. All the groups are produced once in each cycle. Every other group is produced between the production duration of any groups. As the groups are produced only once in each cycle, the production sequence of the groups is not important. We also apply the time-varying lot sizes approach to the items in a group, so that some items may be produced several times during a cycle and their production runs within a cycle may be different giving different lot sizes. Figure 1 shows a group common cycle and time-varying lot sizes approach to the items for two groups which consist of one and two items, assuming that the items are produced in the order 11–12–22–12.

In addition to this, we employ the modified cycle length for balancing between setup and inventory holding costs. The problem requires specifying the production frequency first and then determining the cycle length, the production sequence, idle time, and production time duration. To obtain the production sequence **f**, we modify Kuo and Inman's (1990) sequencing procedure. Dobson (1987) showed that, given

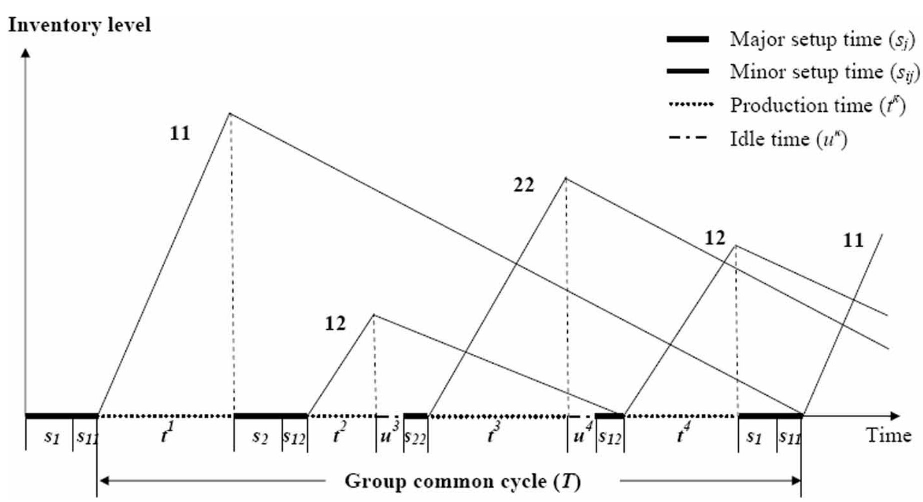

Figure 1. Production schedule for two groups and three items.

a production sequence **f**, there always exists a feasible solution of the problem if and only if $\gamma > 0$.

Our heuristic algorithm to solve the above problem can be described as follows.

Step 1: Set the production frequencies of all items to 1; that is, $m_{ij} = 1$, $\forall i$, $\forall j$.
Step 2: Calculate the cycle length. To ensure that there is enough time in the cycle to admit setups, the cycle length T^* cannot be less than $T_{\min}$ and takes a maximum duration of T' and $T_{\min}$ (Kuo and Inman 1990)

$$T^* = \max\{T', T_{\min}\}, \tag{1}$$

$$T' = \sqrt{\frac{2\sum_{j=1}^{J}(A_j + \sum_{i=1}^{n_j} m_{ij}A_{ij})}{\sum_{j=1}^{J}\sum_{i=1}^{n_j}(H_{ij}/m_{ij})}}, \tag{2}$$

$$T_{\min} = \frac{\sum_{j=1}^{J}(s_j + \sum_{i=1}^{n_j} m_{ij}s_{ij})}{1 - \sum_{j=1}^{J}\sum_{i=1}^{n_j}(d_{ij}/p_{ij})}, \tag{3}$$

where

$$H_{ij} = h_{ij}d_{ij}\left(1 - \frac{d_{ij}}{p_{ij}}\right). \tag{4}$$

Step 3: Generate a production sequence in each group in descending m_{ij} and n_{ij} order. We use the lexicographic order of (m_{ij}, n_{ij}) to assign items to the positions. In each group, all items except $m_{ij} = 0$ are ordered lexicographically by (m_{ij}, n_{ij}) and then all m_{ij} in the group are decreased by 1. Until all m_{ij} in the group are 0, items are ordered lexicographically. Given production frequencies m_{ij}, this sequence procedure attempts to spread out items as evenly as possible. n_{ij} is determined by the following:

$$n_{ij} = \frac{H_{ij}}{A_{ij}}.$$

Note that the item with the larger n_{ij} value is produced earlier than that with the smaller n_{ij} value since we want to assign the item with the larger H_{ij} value first.
Step 4: Solve for the idle time **u** and production time **t** given the production sequence **f** in order to calculate feasible lot sizes and total average cost.

First, if we assume that there is the same idle time except for the first position of each group by dividing the available idle time into equal segments to precede all lots, we can find **u** using equation (5) (Kuo and Inman 1990).

If position k is the first in each group, u^k is 0.

Otherwise,

$$u^k = \left[T - \sum_{j=1}^{J}\left\{s_j + \sum_{i=1}^{n_j}(m_{ij}s_{ij} + Td_{ij}/p_{ij})\right\}\right] \Big/ (l - J). \tag{5}$$

Second, we can find **t** using equation (6) given **f** and **u** (see Dobson (1987) for more details)

$$t = (I - P^{-1}L)^{-1}P^{-1}L(s + u). \tag{6}$$

After calculating the feasible production duration, the total average cost of the schedule is

$$TC = (1/T)\left\{\left[\sum_{j=1}^{J}\left(A_j + \sum_{i=1}^{n_j} m_{ij}A_{ij}\right)\right] + \left[0.5\sum_{k=1}^{l} H_k(t^k p_k/d_k)^2\right]\right\}. \tag{7}$$

Step 5: Calculate α from the total average cost obtained in Step 4. After calculating T^* at the total average cost function under the common cycle approach in Step 2, we apply the time-varying lot sizes approach to the items. The inventory holding cost of the total average cost function has been changed to those in equation (7). Since we use the feasible production duration, t^k, we try to balance between setup and inventory holding costs using α. In order to revise the cycle length T, we set T to αT in equation (7) such that

$$TC = (1/\alpha T)\left\{\left[\sum_{j=1}^{J}\left(A_j + \sum_{i=1}^{n_j} m_{ij}A_{ij}\right)\right] + \alpha^2\left[0.5\sum_{k=1}^{l} H_k(t^k p_k/d_k)^2\right]\right\}. \tag{8}$$

We set $dTC/d\alpha$ to zero and get the ratio α to the cycle length that minimizes the total average costs

$$\alpha = \sqrt{\frac{\sum_{j=1}^{J}(A_j + \sum_{i=1}^{n_j} m_{ij}A_{ij})}{0.5\sum_{k=1}^{l} H_k(t^k p_k/d_k)^2}}. \tag{9}$$

Step 6: Calculate the modified cycle length αT^* as follows:

$$\alpha T^* = \max\{\alpha T', T_{\min}\}. \tag{10}$$

Step 7: Calculate feasible lot sizes and total average cost using αT^* obtained in Step 6. We can find the idle time **u** using αT^* instead of T^* and production time **t** using αT^*, given **f** and **u**. Compute the total average cost using equation (7).

Step 8: If we assume that m_{ij} within a group is satisfied with the practicality rule, we can choose each m_{ij} with the lowest total average cost by changing each m_{ij} given other m_{ij} fixed. Fix all m_{ij} with the lowest total average cost and go to Step 2 until the total average cost is not improved.

Practicality rule. Say that product v is the largest frequency in group j, and product w is the second largest. Then if $m_{vj} > m_{wj} + 1$, set $m_{vj} = m_{wj} + 1$. This rule simply keeps the frequencies of the two most frequent products in each group close (Kuo and Inman 1990).

Note that our heuristic algorithm is different from Kuo and Inman's heuristic in the starting point, Step 1, and terminating conditions, Step 8. They also did not consider α and the modified cycle length αT^* in Steps 5–7, so that their terminating condition is actually Step 5.

4. Hybrid genetic algorithm

This section explains the hybrid genetic algorithm for the GT-ELSP. We use a genetic scheme to find a good production frequency. The main ideas of a genetic algorithm are introduced briefly. How we adapt a genetic algorithm to our problem and why it is a hybrid genetic algorithm will be explained.

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 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, genetic operators, which are reproduction, crossover and mutation, are used for adaptation of an environment and the chromosomes are evaluated by some measures of fitness.

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

- Representation and initialization.
- Penalty function and fitness function.
- Reproduction, crossover and mutation.
- Stopping rule and parameter selection.

4.1 *Representation and initialization*

The proper representation of a solution plays a key role in the development of a genetic algorithm. Traditionally, chromosomes are a simple binary string. This simple representation is not well suited for combinatorial problems, therefore a chromosome consisting of integers is a solution in this paper. The length of the chromosome represents the sum of all item numbers in each group and each gene in the chromosome is the production frequency of each item. For example, suppose we have a five-item, two-group problem having to be produced 1, 2, 1, 1, and 2 times, respectively, during a cycle. Then, we have the following chromosome information and its length is 5 (figure 2). Instead of generating chromosomes purely at random, we apply Kuo and Inman's heuristic solutions to the initial population. An initial population of heuristic solutions is more likely to increase the speed of convergence towards the optimum.

4.2 *Penalty function and fitness function*

The penalty function technique is well known for handling infeasible solutions in constrained optimization problems. The approach converts the constrained problem into an equivalent unconstrained problem. Two basic penalty functions exist: an exterior penalty function and an interior penalty function. In order to find a feasible solution satisfying the constraint of the practicality rule effectively, we use an exterior penalty function which penalizes only infeasible solutions in this research.

1	2	1	1	2

Figure 2. Structure of a chromosome.

The objective is to find a string with the minimum fitness function value. For any given string, we can find the modified cycle length and the production time duration from the heuristic algorithm in section 3.2 and compute the objective function, equation (7). The penalty function added to the objective function value is the fitness function value of the chromosome

$$\begin{aligned} &\text{eval}(x) = f(x) + p(x), \\ &p(x) = 0, \qquad \text{if } x \text{ is feasible}, \\ &p(x) > 0, \qquad \text{otherwise}, \end{aligned}$$

where $f(x)$ is the objective function, $p(x)$ is the penalty function, and x represents a chromosome.

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. These three genetic operators affect the scope of the search.

Reproduction is a process in which individual strings create a new population for the next generation based on their fitness function values. In this paper, parents are chosen with a rank-based mechanism. Unlike other reproduction approaches, a ranking approach offers a smoother selection probability curve. This prevents good organisms from completely dominating the evolution from an early point.

The crossover operation is a diversification mechanism. This diversification mechanism enables the GA to examine unvisited regions. In this study, we use a uniform crossover operator which enables us to exchange bits rather than segments. The uniform crossover is considered better at preserving schema, and can generate any schema from a pair of parents (Michalewicz 1999).

In genetic algorithms, mutation is a background operator, which produces spontaneous random changes in various chromosomes. Mutation serves the crucial role of replacing the genes lost from the population during evolution so that they can maintain diverse populations, thereby preventing the population from being too rigid to adapt to a dynamic environment. We perform mutation by looking at each variable individually. A random number between 0 and 1 is generated for each of the genes in the chromosome. If a gene gets a number that is less than or equal to the mutation rate, then that gene is mutated.

4.4 *Stopping rule and parameter selection*

In this study, we performed computational experiments with the following values for the GA parameters. In order to select appropriate values for the GA parameters, we conducted experiments using 10 randomly generated problems. Three different crossover and mutation rates are considered. Computational results for parameter selection are shown in table 1. It can be seen that the mutation rate is insignificant, whereas the crossover rate is more significant than the mutation rate for the problem. We considered the number of generations up to 1000 even for small-sized problems, but we used tighter termination conditions to reduce the computational effort for large-sized problems. The algorithm stops once either no improvement is made in 50

Table 1. Computational results for parameter selection.

	Mean error (%)		
P_m/P_c	0.4	0.6	0.8
0.2	0.29	0.19	0.33
0.3	0.36	0.20	0.26
0.4	0.24	0.18	0.24

generations or the best individual cannot be improved more than 0.01% over 100 generations

- Population size: 100
- Crossover rate (P_c): 0.6
- Mutation rate (P_m): 0.4.

Now we explain how our hybrid genetic algorithm can improve upon Kuo and Inman's heuristic. Our genetic algorithm uses the modified cycle length rather than the cycle length in Kuo and Inman's heuristic. In addition, we use a genetic scheme to find a good production frequency rather than iterating the heuristic procedure. That is why we call this algorithm a *hybrid* genetic algorithm.

5. Computational experiments

First we solve Ham *et al.*'s (1985) five-item, two-group problem to compare the hybrid genetic algorithm with Kuo and Inman's heuristic. The data for Ham *et al.*'s problem are shown in table 2.

We first show the detailed steps of Kuo and Inman's heuristic as follows.

Step 1: We find the production frequencies satisfied by the practicality rule for every item of each group

$$n_{11} = 0.019, \quad n_{21} = 0.032, \quad n_{12} = 0.009, \quad n_{22} = 0.011, \quad n_{32} = 0.027,$$

and the smallest ratio $\overline{n}_{12} = 0.009$. Round $m_{ij} = \sqrt{n_{ij}/\overline{n}_{ij}}$ to the nearest integers:

$$m_{11} = 1, \quad m_{21} = 2, \quad m_{12} = 1, \quad m_{22} = 1, \quad m_{32} = 2.$$

These frequencies are already satisfied by the practicality rule.

Step 2: Calculate the cycle length (T^*)

$$T' = 58.97, \quad T_{\min} = 21.97,$$
$$T^* = \max\{58.97, 21.97\} = 58.97.$$

Step 3: Generate a production sequence in each group. We use the following lexicographic order of m_{ij} and n_{ij} to assign items to the positions.

Group 1

$$(m_{21}, n_{21}) = (2, 0.032) \geq^L (m_{11}, n_{11}) = (1, 0.019).$$

Table 2. Data for Ham *et al.*'s example.

Group j	Item i	p_{ij}	d_{ij}	h_{ij}	A_{ij}	s_{ij}	A_j	s_j
1	1	150	10	0.010	5	0.9	350	2
	2	200	30	0.010	8	0.2		
2	1	100	15	0.005	7	0.2	400	3
	2	180	25	0.005	10	0.1		
	3	120	20	0.008	5	0.3		

Group 2

$$(m_{32}, n_{32}) = (2, 0.027) \geq^L (m_{22}, n_{22}), = (1, 0.011)$$
$$\geq^L (m_{12}, n_{12}) = (1, 0.009).$$

In the case of group 1, item 2 is first assigned to the first position. Next, item 1 is assigned to the next position. Then, m_{21} is reduced from 2 to 1 and m_{11} is reduced from 1 to 0. Item 2, whose m_{ij} is not zero, is assigned to the next position. m_{21} is reduced to 0. In the case of group 2, item 3 is first assigned to the first position. Item 2 is assigned to the next position and the other is assigned. Then, m_{32} is reduced from 2 to 1. m_{22} and m_{12} are also reduced from 1 to 0. Item 3, whose m_{ij} is not zero, is assigned to the next position. m_{32} is reduced to 0. The resulting production sequence is as follows:

$$\mathbf{f} = (21, 11, 21, 32, 22, 12, 32).$$

Step 4: Compute the idle time $\mathbf{u}$ using equation (5), and production time $\mathbf{t}$ using equation (6) given the production sequence $\mathbf{f}$. Compute the total average daily cost

$$\mathbf{u} = (0, 2.426, 2.426, 0, 2.426, 2.426, 2.426),$$
$$\mathbf{t} = (1.669, 3.932, 7.177, 5.098, 8.191, 8.846, 4.731),$$

with total average daily cost \$28.52.

Step 5: Choose the schedule with the lowest cost by changing m_{ij} gradually. Table 3 shows the iterations finding the schedule with the lower total cost by changing m_{ij} gradually. The best solution is

$$m_{11} = 2, \quad m_{21} = 3, \quad m_{12} = 2, \quad m_{22} = 2, \quad m_{32} = 3,$$

with total average daily cost \$26.48.

Remarks: The above result using Kuo and Inman's heuristic is different from the original one (Kuo and Inman 1990). Kuo and Inman reported \$26.60, whereas our calculation resulted in \$26.48. The heuristic procedure is the same. However, the portions of production time $\mathbf{t}$ in a group obtained in Step 4 are different.

We show the detailed steps of our heuristic using the same example as follows.

Step 1: Set $m_{ij} = 1$, $\forall i$, $\forall j$

$$m_{11} = 1, \quad m_{21} = 1, \quad m_{12} = 1, \quad m_{22} = 1, \quad m_{32} = 1.$$

Table 3. Total average cost comparison by changing m_{ij}.

	m_{11}	m_{21}	m_{12}	m_{22}	m_{32}	T (days)	TC ($)
Add 1 to all m_{ij}	1	2	1	1	2	58.97	28.52
	2	3	2	2	3	79.77	26.48
	3	4	3	3	4	96.79	26.81
Decrease the largest m_{ij} by 1	3	3	3	3	4	91.25	27.18
	3	3	3	3	3	88.63	27.34
	2	3	3	3	3	85.37	26.75
	2	2	3	3	3	78.15	27.18
	2	2	2	3	3	76.37	26.81
	2	2	2	2	3	73.64	27.11
	2	2	2	2	2	70.87	27.56
	1	2	2	2	2	66.09	27.29
	1	1	2	2	2	56.78	29.02
	1	1	1	2	2	54.81	29.25
	1	1	1	1	2	51.91	30.45
	1	1	1	1	1	49.03	32.02

Step 2: Calculate the cycle length T^*

$$T' = 49.03, \quad T_{\min} = 20.44,$$
$$T^* = \max\{49.03, 20.44\} = 49.03.$$

Step 3: Generate a production sequence in each group. We use the following lexicographic order of (m_{ij}, n_{ij}) to assign items to the positions:

$$\mathbf{f} = (21, 11, 32, 22, 12).$$

Step 4: Compute the idle time **u** using equation (5) and production time **t** using equation (6). Compute the total average daily cost using equation (7)

$$\mathbf{u} = (0, 3.124, 0, 3.124, 3.124),$$
$$\mathbf{t} = (7.355, 3.269, 8.172, 6.810, 7.355),$$

with total average daily cost $32.02.

Step 5: Calculate α from the total average cost obtained in Step 4

$$\alpha = \sqrt{\frac{875}{875}} = 1.$$

Steps 6 and 7: Because α is 1 at the first iteration, the modified cycle length αT^* and the idle time **u**, production time **t**, and total average cost are equal to the results in Step 2 and Step 4.

Step 8: If we assume that m_{ij} within a group is satisfied by the practicality rule, we can choose each m_{ij} with the lowest total average cost by changing each m_{ij}. Fix all m_{ij} with the lowest total average cost and go to Step 2 until the total average cost is not improved.

The results of several iterations are shown in table 4. The best solution is

$$m_{11} = 2, \quad m_{21} = 3, \quad m_{12} = 2, \quad m_{22} = 3, \quad m_{32} = 4,$$

Table 4. Total average cost comparison by fixing m_{ij} with the lowest cost.

Iteration	m_{11}	m_{21}	m_{12}	m_{22}	m_{32}	T^* (days)	αT^* (days)	TC ($)
1	1	1	1	1	1	49.03	49.03	32.02
2	2	2	2	2	2	70.87	60.29	27.17
3	2	3	2	3	3	83.15	65.95	25.55
4	2	3	2	3	4	85.36	66.46	25.51

with total average daily cost $25.51. Our heuristic improved over Kuo and Inman's heuristic by 4.10%.

We also solve the same example, but this time using a genetic algorithm. Figure 3 shows the structure of our genetic algorithm and the detailed steps of the genetic algorithm are shown below.

Step 1: We find the production frequencies using the genetic algorithm

$$m_{11} = 6, \quad m_{21} = 7, \quad m_{12} = 2, \quad m_{22} = 3, \quad m_{32} = 4.$$

Step 2: Calculate the cycle length T^*

$$T' = 108.44, \quad T_{\min} = 41.80,$$
$$T^* = \max\{108.44, 41.80\} = 108.44.$$

Step 3: Generate a production sequence in each group. We use the following lexicographic order to assign items to the positions

Group 1

$$(m_{21}, n_{21}) = (7, 0.032) \geq^L (m_{11}, n_{11}) = (6, 0.019).$$

Group 2

$$(m_{32}, n_{32}) = (4, 0.027) \geq^L (m_{22}, n_{22}) = (3, 0.011)$$
$$\geq^L (m_{12}, n_{12}) = (2, 0.009).$$

The resulting production sequence is as follows:

$$\mathbf{f} = (21, 11, 21, 11, 21, 11, 21, 11, 21, 11, 21, 11, 21, 32, 22, 12, 32, 22, 12, 32, 22, 32).$$

Step 4: Compute the idle time $\mathbf{u}$ using equation (5) and production time $\mathbf{t}$ using equation (6). Compute the total average daily cost using equation (7)

$$\mathbf{u} = (0, 1.092, 1.092, 1.092, 1.092, 1.092, 1.092, 1.092, 1.092, 1.092, 1.092, 1.092, 1.092, 0, 1.092, 1.092, 1.092, 1.092, 1.092, 1.092, 1.092, 1.092),$$
$$\mathbf{t} = (0.79, 0.28, 0.63, 0.28, 0.63, 0.28, 0.63, 0.28, 0.64, 0.35, 1.60, 5.76, 11.35, 1.89, 1.64, 2.03, 4.29, 3.33, 14.24, 2.53, 10.09, 9.36),$$

with total average daily cost $26.78.

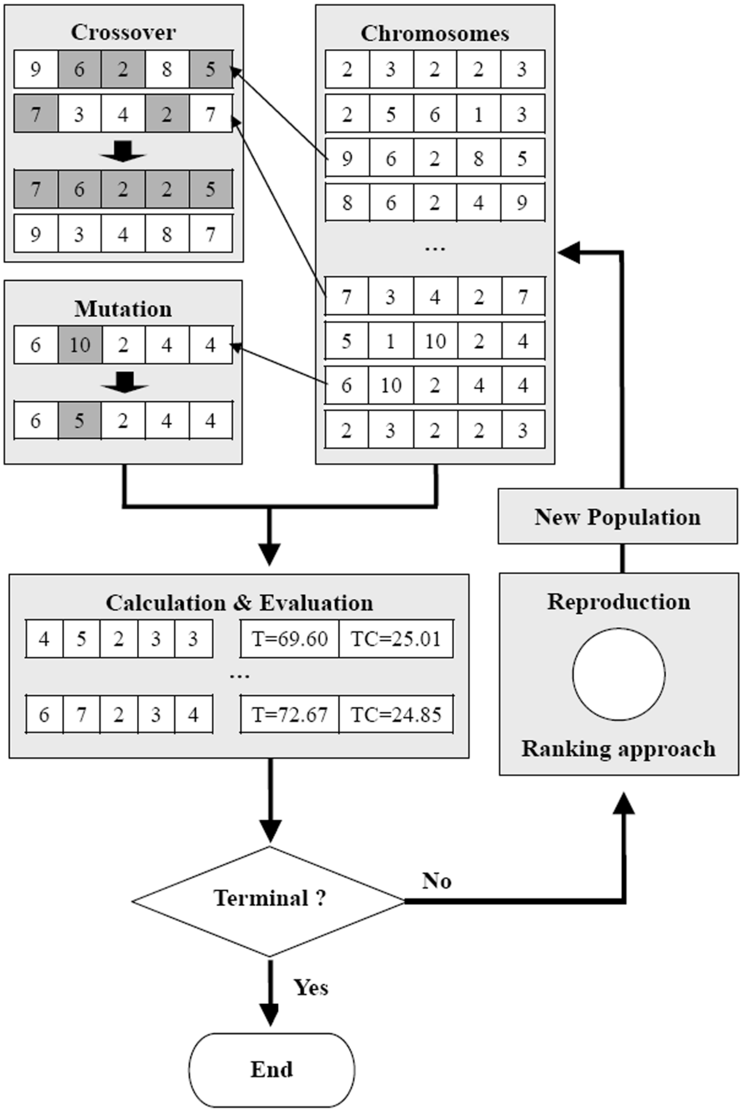

Figure 3. The structure of the proposed genetic algorithm.

Step 5: Calculate α from the total average cost obtained in Step 4

$$\alpha = \sqrt{\frac{900}{2004}} = 0.67.$$

Step 6: Calculate the modified cycle length αT^*

$$\alpha T' = 72.67, \quad T_{\min} = 41.80,$$
$$\alpha T^* = \max\{72.67, 41.80\} = 72.67.$$

Step 7: Calculate feasible lot sizes and total average cost using αT^* obtained in Step 6. We can find the idle time $\mathbf{u}$ using αT^* instead of T^* and production time $\mathbf{t}$ using αT^*, given $\mathbf{f}$ and $\mathbf{u}$. Compute the total average cost using equation (7)

$$\mathbf{u} = (0, 0.506, 0.506, 0.506, 0.506, 0.506, 0.506, 0.506, 0.506, 0.506, 0.506, 0.506, 0.506, 0, 0.506, 0.506, 0.506, 0.506, 0.506, 0.506, 0.506, 0.506),$$

$$\mathbf{t} = (0.67, 0.18, 0.40, 0.18, 0.40, 0.18, 0.40, 0.18, 0.41, 0.23, 1.06, 3.90, 7.55, 1.37, 0.99, 1.25, 2.79, 2.17, 9.65, 1.67, 6.93, 6.28),$$

with total average daily cost \$24.85. We have improved over Kuo and Inman's heuristic by 6.58%.

The computational results for Ham *et al.*'s example are summarized in table 5. It shows significant savings with respect to Kuo and Inman's heuristic.

Table 5. Results for Ham *et al.*'s problem.

Algorithm	m_{11}	m_{21}	m_{12}	m_{22}	m_{32}	T (days)	TC (\$)	Saving
Kuo and Inman's heuristic	2	3	2	2	3	79.76	26.60	
Our heuristic	2	3	2	3	4	66.46	25.51	4.10%
Hybrid GA	6	7	2	3	4	72.67	24.85	6.58%

Table 6. Distributions for randomly generated data for test problems.

Parameter		Interval
Major setup time (unit times)	s_j	[1, 3]
Major setup cost (\$)	A_j	[300, 500]
Minor setup time (unit times)	s_{ij}	[0.1, 0.3]
Minor setup cost (\$)	A_{ij}	[5, 10]
Production rate (units/unit time)	p_{ij}	See table 7
Demand rate (units/unit time)	d_{ij}	See table 7
Holding cost (\$/unit/unit time)	h_{ij}	[0.005, 0.01]

Table 7. Uniform distributions of production and demand rate.

Group	Item	Utilization		Production rate interval	Demand rate interval
2	5	Low	[0.3, 0.5]	[200, 300]	
		Medium	[0.5, 0.7]	[150, 250]	[10, 30]
		High	[0.7, 0.9]	[100, 200]	
3	15	Low	[0.3, 0.5]	[300, 400]	
		Medium	[0.5, 0.7]	[200, 300]	[5, 15]
		High	[0.7, 0.9]	[150, 250]	
5	25	Low	[0.3, 0.5]	[350, 500]	
		Medium	[0.5, 0.7]	[250, 400]	[5, 10]
		High	[0.7, 0.9]	[150, 300]	

Table 8. Comparison of four algorithms (% improvement).

Group/Item	Utilization	Number of best solution problems		Our heuristic better than Ham *et al.*		Our heuristic better than Kuo and Inman		Hybrid GA better than Ham *et al.*		Hybrid GA better than Kuo and Inman	
		Our heuristic	Hybrid GA	Max.	Ave.	Max.	Ave.	Max.	Ave.	Max.	Ave.
2/5	Low	3	27	36.29	28.82	9.45	5.98	36.33	29.35	9.51	6.68
	Medium	1	29	30.78	26.48	7.91	5.93	31.31	27.02	8.88	6.63
	High	2	28	28.23	23.78	7.94	5.22	29.29	24.16	8.16	5.71
3/15	Low	6	24	13.98	12.42	4.02	3.03	14.17	12.60	4.51	3.23
	Medium	5	25	13.14	11.34	3.48	2.33	13.61	11.51	3.73	2.52
	High	4	26	12.84	9.86	3.50	1.03	12.79	10.84	3.96	2.10
5/25	Low	0	30	5.49	4.57	1.69	0.64	6.35	5.15	2.33	1.25
	Medium	0	30	4.99	4.32	2.03	0.62	5.66	5.06	2.70	1.39
	High	1	29	3.60	1.74	2.61	0.05	4.91	3.27	3.31	1.61
Max.				36.29		9.45		36.33		9.51	
Ave.		2.44	27.56		13.70		2.76		14.33		3.46

In addition, we performed computational experiments to evaluate the performance of our heuristic and the hybrid genetic algorithm. Three test problems of different size are considered, each test problem consisting of three sets of problems corresponding to the different utilization of low, medium, and high. In each test problem, we compared the performance of four algorithms on three sets of 30 randomly generated problems from uniform distributions on the given intervals (see table 6). In order to obtain the utilization in the proper range, we divided the distribution of the utilization into three ranges as shown in table 7. Note that the test problem that consists of two groups and five items was generated from the same distributions as in Kuo and Inman (1990). We solved each test problem using Ham *et al.*'s (1985) algorithm, Kuo and Inman's (1990) heuristic, our heuristic and the hybrid genetic algorithm. Table 8 displays the computational results of the four algorithms for three different sized problems. The number of best solution problems presented in table 8 is the number of times the proposed algorithms produced the best solution among the four algorithms. Based on this result, the hybrid genetic algorithm gives the best solution for the most part for different problem sizes and utilization. The improvement of our hybrid genetic algorithm over Kuo and Inman's heuristic is Kuo and Inman's cost minus the cost of the hybrid genetic algorithm, divided by Kuo and Inman's cost. The computational study demonstrates that the hybrid genetic algorithm outperforms Kuo and Inman's heuristic algorithm. However, the improvement rate decreases as the problem size and the utilization rate increase.

6. Conclusions

The Economic Lot Scheduling Problem (ELSP) is an important production scheduling problem that has been studied intensively for over 40 years. Numerous heuristic algorithms have been developed, since the problem is NP-hard. This paper considers the use of group technology principles in the economic lot scheduling problem. One of the most prominent advantages of applying group technology to production is reduction of the setup cost and time, which play a critical role in production planning. In this paper, we develop a hybrid genetic algorithm for the Group Technology Economic Lot Scheduling Problem (GT-ELSP). We have also carried out numerical experiments for the GT-ELSP, and our hybrid genetic algorithm outperformed Kuo and Inman's heuristic. In this paper, we have only applied the genetic schema to the production frequency. However, genetic algorithms may be applied to both the production frequency and sequence simultaneously. Another purpose of developing a heuristic algorithm is to solve the ELSP for each group independently, and then try to concatenate the obtained sequences in a way that minimizes the total cost.

Acknowledgements

This work was supported by the Regional Research Centers Program (Research Center for Logistics Information Technology) of the Korean Ministry of Education and Human Resources Development.

References

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

Burbidge, J., Operation scheduling with GT and PBC. *International Journal of Production Research*, 1988, **26**, 429–442.

Delporte, C. and Thomas, L., Lot sizing and sequencing for *N* products on one facility. *Management Science*, 1977, **23**, 1070–1079.

Dobson, G., The economic lot-scheduling problem: achieving feasibility using time-varying lot sizes. *Operations Research*, 1987, **35**, 764–771.

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

Gallego, G., An extension to the class of easy economic lot scheduling problems. *IIE Transactions*, 1990, **22**, 189–190.

Gallego, G. and Moon, I., The effect of externalizing setups in the economic lot scheduling problem. *Operations Research*, 1992, **40**, 614–619.

Gallego, G. and Shaw, D., Complexity of the ELSP with general cyclic schedules. *IIE Transactions*, 1997, **29**, 109–113.

Ham, I., Hitomy, K. and Yoshida, T., *Group Technology: Applications to Production Management*, 1985 (Kluwer–Nijhoff: Boston, MA).

Hanssmann, F., *Operations Research in Production and Inventory Control*, 1962 (Wiley: New York).

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

Jones, P. and Inman, R., When is the economic lot scheduling problem easy? *IIE Transactions*, 1989, **21**, 11–20.

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

Kim, M. and Lee, H., A study on the capacitated production lot sizing problem with probabilistic demand. *Journal of the Korean Institute of Industrial Engineers*, 1989, **15**, 109–116.

Kuo, H. and Inman, R., A practical heuristic for the group technology economic lot scheduling problem. *International Journal of Production Research*, 1990, **28**, 709–722.

Maxwell, W., The scheduling of economic lot sizes. *Naval Research Logistics Quarterly*, 1964, **11**, 89–124.

Michalewicz, Z., *Genetic Algorithms+Data Structures=Evolution Programs*, 3rd ed., 1999 (Springer: New York).

Moon, I., Hahm, J. and Lee, C., The effect of the stabilization period on the economic lot scheduling problem. *IIE Transactions*, 1998, **30**, 1009–1017.

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

Opitz, H., *A Classification System to Describe Workpieces: Parts I and II*, 1970 (Pergamon: Oxford, UK).

Ree, P., A heuristic algorithm for forming machine cells and part families in group technology. *Journal of the Korean Institute of Industrial Engineers*, 1996, **22**, 705–718.

Silver, E., Pyke, D. and Peterson, R., *Inventory Management and Production Planning and Scheduling*, 3rd ed., 1998 (Wiley: New York).

Wemmerlöv, U. and Hyer, N., Cellular manufacturing in the U.S. industry: a survey of users. *International Journal of Production Research*, 1989, **27**, 1511–1530.

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