International Journal of Production Research,
Vol. 46, No. 10, 15 May 2008, 2695–2705

Genetic algorithms for job shop scheduling problems with alternative routings

ILKYEONG MOON*†, SANGHYUP LEE‡ and HYERIM BAE†

†Department of Industrial Engineering, Pusan National University, Busan 609-735, Korea
‡Hyundai Industrial Research Institute, Hyundai Heavy Industries Company, Korea

(*Revision received January 2007*)

The purpose of this research is to solve a general job shop problem with alternative machine routings. We consider four performance measures: mean flow time, makespan, maximum lateness, and total absolute deviation from the due dates. We first develop mixed-integer linear programming (MILP) formulations for the problems. The MILP formulations can be used either to compute optimal solutions for small-sized problems or to test the performance of existing heuristic algorithms. In addition, we have developed a genetic algorithm that can be used to generate relatively good solutions quickly. Further, computational experiments have been performed to compare the solution of the MILP formulations with that of existing algorithms.

Keywords: Job shop; Production scheduling; FMS; Genetic algorithm

1. Introduction

Process planning and production scheduling are two of the most important functions of a manufacturing system. In process planning, each machining operation is assigned to a certain machine tool, whereas in production scheduling, each machine is assigned different operations over time. Even though process planning and production scheduling are highly related, we often notice that these two functions are performed separately. The separation of process planning and production scheduling results in many problems due to conflicting objectives or the inability to communicate the dynamic changes in the job shop. As a result, a production schedule that lacks flexibility and adaptability is produced (Nasr and Elsayed 1990). In this study, we emphasize machine assignment and task scheduling in an integrated system to overcome these problems. In addition, we make the system more flexible by allowing alternative machine tool routings for the operations. An alternative operation could be used if one machine tool is temporarily overloaded

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

International Journal of Production Research
ISSN 0020–7543 print/ISSN 1366–588X online © 2008 Taylor & Francis
http://www.tandf.co.uk/journals
DOI: 10.1080/00207540701244820

while another is idle. Even though alternative operations may incur time penalties, they may be used to offload bottleneck machines with the objective of balancing machine utilization and expediting the flow of workpieces. In this system, the objective is to simultaneously optimize the assignment of machines for each part process and the loading sequences of parts to machines.

Wilhelm and Shin (1985) conducted a study to investigate the effectiveness of alternative operations in flexible manufacturing systems. Through computational experiments, they showed that alternative operations can reduce the flow time while increasing machine utilization. Recently, Nasr and Elsayed (1990) developed an efficient heuristic to minimize the mean flow time in a general job shop type machining system with alternative machine tool routings. However, they did not provide an algorithm to solve the problem optimally. In this paper, we develop a mixed-integer linear programming formulation (MILP) for the problem to complement the work of Nasr and Elsayed (1990). In addition to the basic formulation, we show that three other performance measures, i.e. makespan, maximum lateness, and total absolute deviation from the due dates, can easily be accommodated by slightly modifying the basic formulation. Since MILP cannot be used to solve large-sized problems, we develop a genetic algorithm for solving these problems.

Several past studies have adopted a mathematical programming technique to solve scheduling and assignment problems that arise in manufacturing systems. Liang and Dutta (1990) developed a mixed-integer programming formulation to simultaneously consider the problems of process planning and machine loading. This formulation has been used to provide an optimal process plan for each part and an optimal load for each machine by including the machining cost, material handling cost, setup cost, and machine idle cost. Hutchison *et al.* (1991) provided an optimal solution procedure to investigate the effect of routing flexibility on job shop flexible manufacturing systems (FMS).

Genetic algorithms have proven to be effective for job shop scheduling problems. Many of the studies that have used genetic algorithms for job shop scheduling problems have been summarized by Gen and Cheng (1997). Park *et al.* (1998) applied the genetic algorithm to a job shop system with alternative process plans. Zhou *et al.* (2001) developed a hybrid genetic algorithm for job shop scheduling problems. They applied priority rules such as the shortest processing time for the genetic search to devise a hybrid genetic algorithm. Kim *et al.* (2003) presented a new evolutionary algorithm to solve FMS loading problems with machine, tool, and process flexibilities. Kim *et al.* (2004) introduced an asymmetric multi-leveled symbiotic evolutionary algorithm, and applied it to the integrated problem of process planning and scheduling in FMS.

The organization of the paper is as follows. We first address the MILP formulation to minimize the mean flow time in section 2. Applications of the formulation to other types of performance measures are also explained. A genetic algorithm is developed in section 3. In section 4, we compare the optimal solution obtained from MILP and the solution obtained from our genetic algorithm with the solution of Nasr and Elsayed (1990), and show that additional savings are possible. In addition, computational experiments are performed to analyse the effectiveness of the genetic algorithm. We conclude in section 5.

2. Mixed-integer linear programming formulation

We first introduce the following notation to formulate the problem mathematically.

i	index for job ($i = 1, 2, \dots, I$)
j	index for operation ($j = 1, 2, \dots, J_i$)
k	index for machine ($k = 1, 2, \dots, K$)
p	index for the order of assignment to machine k ($p = 1, 2, \dots, N_k$)
J_i	number of operations required to complete job i
$G(k)$	set of operations that can be assigned to machine k
N_k	number of operations that belong to set $G(k)$
O_{ij}	operation number j of job i
S_{ij}	starting time of operation O_{ij}
F_{ij}	finishing time of operation O_{ij}
R_{kp}	release time of machine k after processing its pth operation in $G(k)$
t_{ijk}	machining time of operation O_{ij} on machine k
r_i	time at which job i becomes ready for processing
H	a very large positive number
X_{ijkp}	$= 1$ if O_{ij} has been assigned to machine k on the pth order; $= 0$ otherwise.

The system consists of K machines and I different jobs. All jobs are processed in a predetermined technological order given in the process plan. J_i operations are required to finish job i. Each of these operations can be processed on a number of alternative, non-identical machines. These assumptions are the same as those used by Nasr and Elsayed (1990).

(1) The jobs are independent and consist of strictly ordered operation sequences. No priorities are assigned to any job or operation.
(2) Job pre-emption is not allowed.
(3) A given operation can be performed by one or more non-identical machines (called alternative machines).
(4) The setup times are independent of the operation sequence and are included in the processing times.

We developed the following MILP formulation to minimize the mean flow time of jobs:

$$\text{Min} \sum_{i=1}^{I} F_{iJ_i},$$

subject to

$$S_{iJ_i} + \sum_{\{k:O_{iJ_i} \in G(k)\}} \sum_{p=1}^{N_k} t_{iJ_ik} X_{iJ_ikp} = F_{iJ_i}, \quad \text{for all } i, \tag{1}$$

$$S_{ij} + \sum_{\{k:O_{ij} \in G(k)\}} \sum_{p=1}^{N_k} t_{ijk} X_{ijkp} \le S_{ij+1}, \quad \text{for all } i, j = 1, 2, \dots, J_i - 1, \tag{2}$$

$$r_i \le S_{i1}, \quad \text{for all } i, \tag{3}$$

$$S_{ij} + (t_{ijk} + H)X_{ijkp} - R_{kp} \le H, \quad \{(i,j) : O_{ij} \in G(k)\},\ p = 1, \dots, N_k - 1, \quad \text{for all } k, \tag{4}$$

$$R_{kp} + HX_{ijkp+1} - S_{ij} \le H, \quad \{(i,j) : O_{ij} \in G(k)\},\ p = 1, \dots, N_k - 1, \quad \text{for all } k, \tag{5}$$

$$R_{kp} - R_{kp+1} \le 0,\ p = 1, 2, \dots, N_k - 2, \quad \text{for all } k, \tag{6}$$

$$\sum_{\{(i,j):O_{ij}\in G(k)\}} X_{ijkp} - \sum_{\{(i,j):O_{ij}\in G(k)\}} X_{ijkp+1} \ge 0, \quad p = 1, 2, \dots, N_k - 1, \quad \text{for all } k, \tag{7}$$

$$\sum_{\{k:O_{ij}\in G(k)\}} \sum_{p=1}^{N_k} X_{ijkp} - \sum_{\{k:O_{ij+1}\in G(k)\}} \sum_{p=1}^{N_k} X_{ij+1kp} \ge 0, \quad \text{for all } i, j = 1, \dots, J_i - 1, \tag{8}$$

$$\sum_{\{(i,j):O_{ij}\in G(k)\}} X_{ijkp} \le 1, \quad \text{for all } k, p = 1, 2, \dots, N_k, \tag{9}$$

$$\sum_{\{k:O_{ij}\in G(k)\}} \sum_{p=1}^{N_k} X_{ijkp} = 1, \quad \text{for all } i, j = 1, 2, \dots, J_i, \tag{10}$$

$$X_{ijkp} = 0 \text{ or } 1, \quad \text{for all } i, j, k, p. \tag{11}$$

The meanings of the constraints are as follows. Constraint (1) represents the flow time of each job i. Constraint (2) implies that the $(j+1)$th operation of a job can only be started after the jth operation has been completed. The starting time of the first operation of each job must be greater than or equal to its ready time, and this is represented by constraint (3). The relationship between the machine release time and the operation assigned to the machine is represented by constraint (4). Note that the constraint becomes inactive if $X_{ijkp} = 0$. Constraint (5) implies that the starting time of an operation that has been assigned to machine k in the $(p+1)$th order must be greater than or equal to the release time of the machine after the completion of its pth operation. The release time of machine k after the completion of its $(p+1)$th operation must be greater than or equal to that after its pth operation; this is represented by constraint (6). Constraint (7) implies that the $(p+1)$th operation that belongs to $G(k)$ can be assigned only after the pth operation has been assigned to machine k. Constraint (8) implies that the $(j+1)$th operation of a job can only be assigned after the pth operation has been assigned to a machine. The assumption that no more than one operation can be assigned to a machine simultaneously is represented by constraint (9). Constraint (10) implies that an operation can only be assigned to a machine.

We now explain several different objective functions and changes in the constraints to represent different performance measures of the manufacturing system. Suppose we want to minimize the makespan; we can change the previous formulation as follows where Z denotes the maximum among all the flow times:

$$\begin{aligned} &\text{Min } Z, \\ &\text{subject to (1)–(11)}, \\ &F_{iJ_i \le Z}, \quad \text{for all } i. \end{aligned} \tag{12}$$

In a just-in-time (JIT) production environment, jobs that are completed early must be held in the finished goods inventory until their due dates, whereas jobs that are completed after their due dates may cause a customer to terminate operations. Therefore, an ideal schedule is one in which all the jobs finish exactly on their assigned due dates. Refer to Baker and Scudder (1990) and Moon and Yun (1997) for further justification for using earliness and tardiness penalties in scheduling problems. One obvious objective function to accommodate this situation is to minimize the total absolute deviation of meeting the due dates. The following formulation can be used to reflect this objective function. Let D_i denote the due date for job i and w_i the weight for job i. O_i is a variable indicating the amount of tardiness of job i, and U_i is a variable indicating the amount of earliness of job i. Note that $O_iU_i = 0$ will be clearly satisfied at the optimal solution

$$\text{Min} \sum_{i=1}^{I} w_i(O_i + U_i),$$

subject to (2)–(11),

$$S_{iJ_i} + \sum_{\{k:O_{iJ_i} \in G(k)\}} \sum_{p=1}^{N_k} t_{iJ_ik} X_{iJ_ikp} - O_i + U_i = D_i, \quad \text{for all } i. \tag{13}$$

In order to minimize the maximum lateness, the following formulation can be utilized:

$$\text{Min } L_{\max},$$

subject to (2)–(11),

$$S_{iJ_i} + \sum_{\{k:\, O_{iJ_i} \in G(k)\}} \sum_{p=1}^{N_k} t_{iJ_ik} X_{iJ_ikp} - L_i = D_i, \quad \text{for all } i, \tag{14}$$

$$L_i \le L_{\max}, \quad \text{for all } i. \tag{15}$$

Note that we need to impose $L_{\max} \geq 0$ on the above constraints in order to minimize the maximum tardiness instead of the maximum lateness.

3. Genetic algorithm

In this section, we present a genetic algorithm approach for solving the job shop scheduling problem with alternative machines. The main ideas of the genetic algorithm are introduced and we demonstrate how the genetic algorithm can be applied to our problem. Genetic algorithms, which have been widely used to solve operations management problems during the last decade, are stochastic search algorithms based on the mechanism of natural selection and natural genetics. Genetic algorithms, which differ from conventional search techniques, start with an initial set of (random) solutions called a population. Each individual in the population is called a chromosome, which represents 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 that are too large to be exhaustively searched. It is generally accepted that any genetic algorithm must have the basic components in order to solve a problem; however, it should have different characteristics depending on the problem under study.

The proper representation of a solution plays a key role in the development of a genetic algorithm. In our genetic algorithm, we can search the assignment of each operation to each machine through operations of the genetic algorithm. Our chromosome is composed of two parts. The first part is for the assignment of alternative machines, and the second part is the relative processing order between jobs (see figure 1). The length of each chromosome is equal to the total number of operations, that is $\sum_{i=1}^{I} J_i$.

To illustrate the chromosome, we use the data in table 1, where there are two jobs and six alternative machines. Suppose a chromosome is given as shown in figure 2. O_{11} is 1 (in the assignment chromosome), which means that the first operation of job 1 is assigned to machine 1, and O_{12} is 5, which means that the second operation of job 1 is assigned to machine 5. The genes that belong to the same job (in the sequence chromosome) must be converted before assigning the jobs to the machines. The conversion of the orders on each job to an increasing order (shown in figure 2) guarantees a feasible solution. $O_{13} = 4$ and $O_{21} = 2$ imply that O_{21} is assigned second,

Assignment chromosome

O_{11} ··· O_{1J_1}	···	O_{I1} ··· O_{1J_1}

Sequence chromosome

O_{11} ··· O_{1J_1}	···	O_{I1} ··· O_{1J_1}

Figure 1. Structure of the chromosome.

Table 1. Example data.

		Alternative machines					
		1	2	3	4	5	6
Job 1	O_{11}	2	3	4	–	–	–
	O_{12}	–	3	–	2	4	–
	O_{13}	1	4	5	–	–	–
Job 2	O_{21}	3	–	5	–	2	–
	O_{22}	4	3	–	–	6	–
	O_{23}	–	–	4	–	7	11

Assignment chromosome

O_{11}	O_{12}	O_{13}	O_{21}	O_{22}	O_{23}
1	5	3	3	2	5

Sequence chromosome

O_{11}	O_{12}	O_{13}	O_{21}	O_{22}	O_{23}
3	4	1	2	6	5

Convert to a feasible solution

O_{11}	O_{12}	O_{13}	O_{21}	O_{22}	O_{23}
1	3	4	2	5	6

Figure 2. Example chromosome.

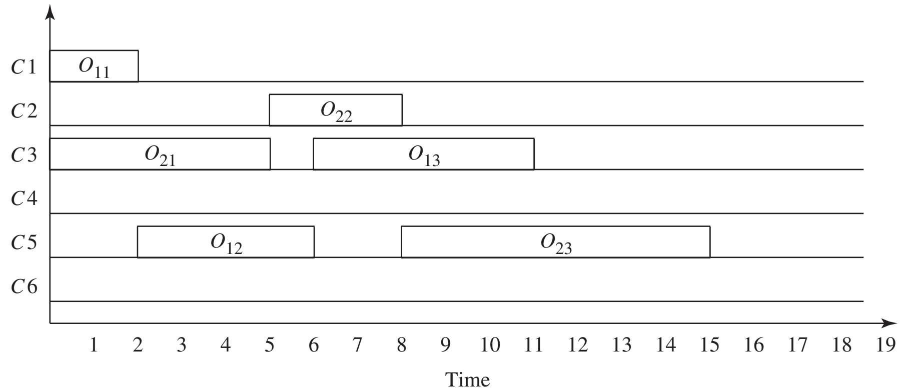

Figure 3. Assignment of operations using the sample chromosome.

and O_{13} is assigned fourth, which implies that O_{21} should be assigned prior to O_{13} on machine 3. The chromosome can be converted into a feasible schedule, as shown in figure 3.

We use a one-point crossover for the chromosome, which is used to assign jobs to alternative machines, and PMX (partial matched crossover) for the chromosome, which decides the relative processing order between the jobs. The structure of the genetic algorithm is depicted in figure 4 using the example explained above. Other genetic algorithm parameters will be explained in the section describing the computational experiments.

4. Numerical examples and computational experiments

We use the same example that is given in Nasr and Elsayed (1990). There are four jobs, and each has three different operations to be processed according to a given sequence (O_{i1}, O_{i2}, O_{i3}). There are six different machines in the system, and the alternative routings and processing times are shown in table 2. We compare the results of the heuristic procedure of Nasr and Elsayed (1990) with those of our MILP and the genetic algorithm. The flow times in their procedure are $F = (7, 11, 18, 13)$, where F_i denotes the flow time of job i. Note that the mean flow time is 12.25. The flow times of the MILP procedure are $F = (6, 11, 17, 13)$, and the mean flow time is 11.75, which is an optimal solution. The genetic algorithm produces the same solution even though the assignment of operations is different from that in the MILP. The detailed assignment of operations to machines is depicted in figure 5.

Remark 1: If we solve the example with the objective of minimizing the makespan, an optimal makespan of 17 is obtained. Note that the makespan computed by Nasr and Elsayed (1990) is 18.

We performed computational experiments to compare the performance of our genetic algorithm with that of Nasr and Elsayed's (1990) heuristic. The data set

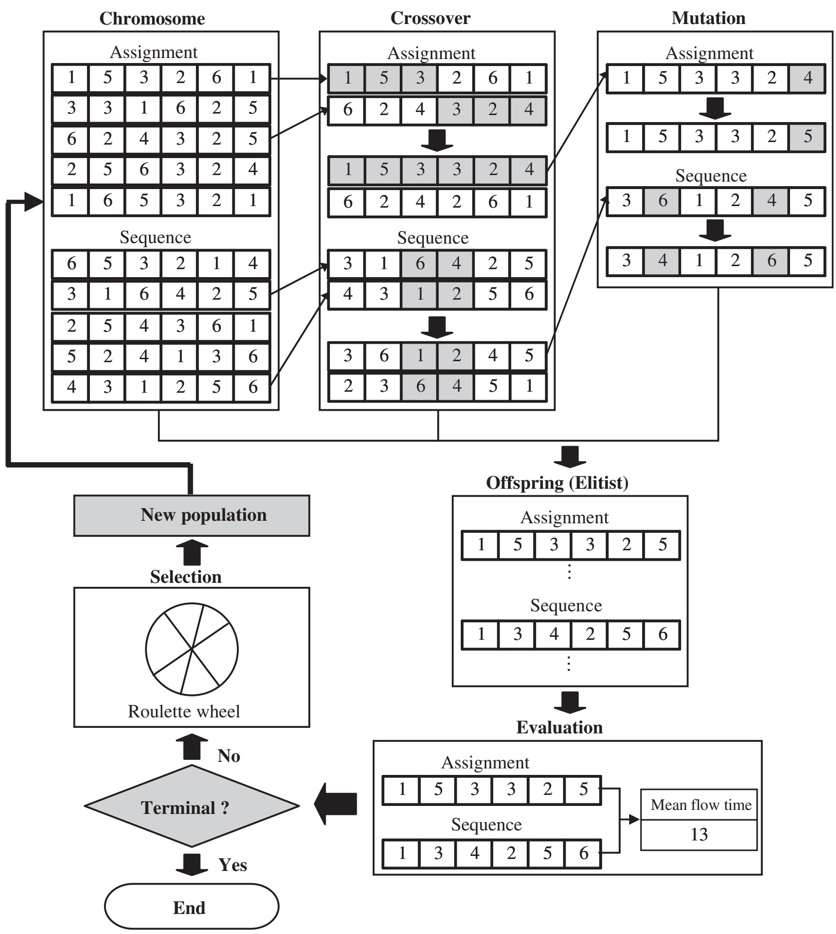

Figure 4. Structure of the genetic algorithm.

Table 2. Problem data.

		Alternative machines					
		1	2		4	5	6
Job 1	O_{11}	2	3	4	–	–	–
	O_{12}	–	3	–	2	4	–
	O_{13}	1	4	5	–	–	–
Job 2	O_{21}	3	–	5	–	2	–
	O_{22}	4	3	–	–	6	–
	O_{23}	–	–	4	–	7	11
Job 3	O_{31}	5	6	–	–	–	–
	O_{32}	–	4	–	3	5	–
	O_{33}	–	–	13	–	9	12
Job 4	O_{41}	9	–	7	9	–	–
	O_{42}	–	6	–	4	–	5
	O_{43}	1	–	3	–	–	3

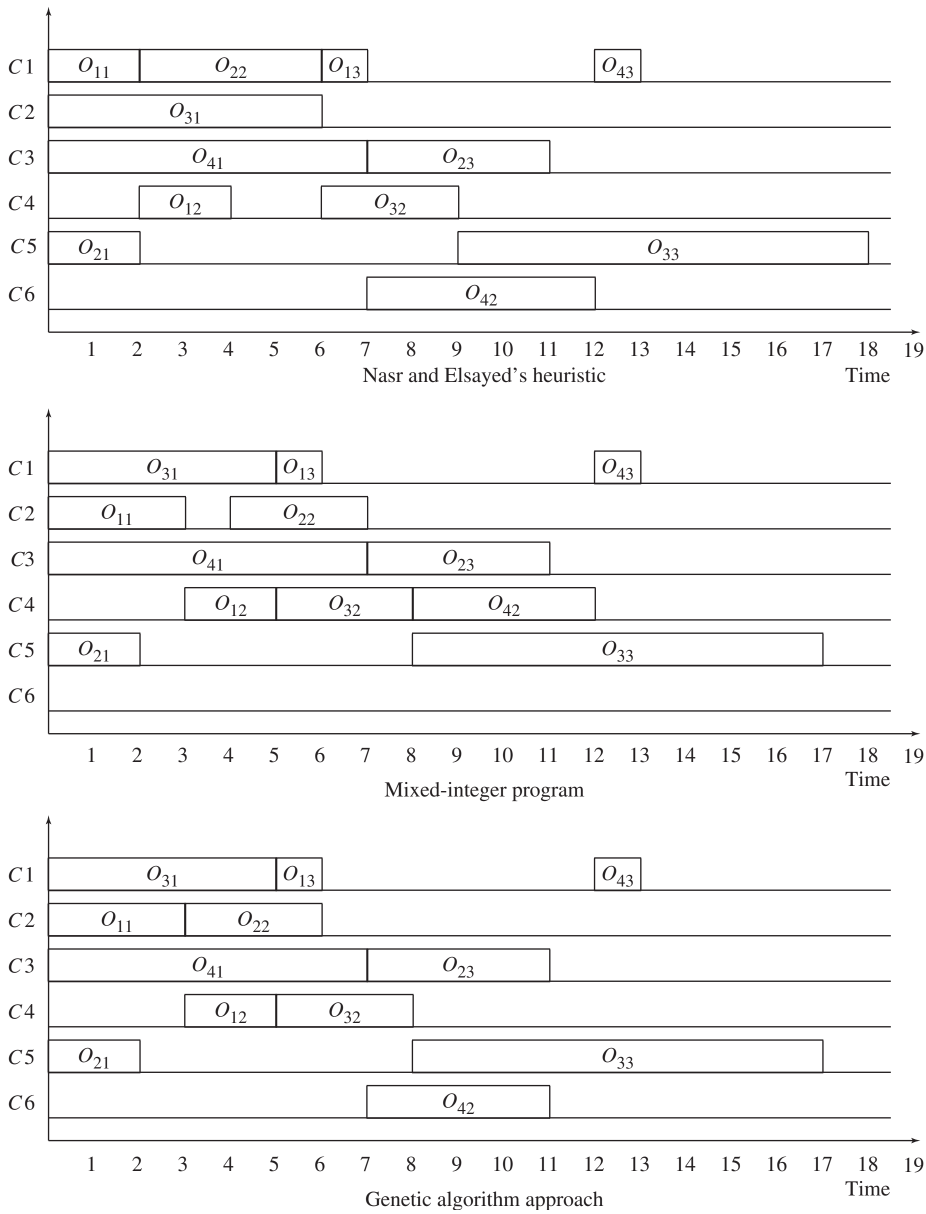

Figure 5. Comparison of the three procedures.

was generated randomly from uniform distributions in the given intervals (see table 3), and 100 problems were generated. Two types of parameters are involved in the computation, namely the problem data and the genetic algorithm parameters. The former includes the number of jobs, number of operations per job, number of machines, etc. The second set of parameters includes the population size, replacement policy, number of generations, crossover rate, and mutation rate.

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

Parameter	Set
Number of jobs	[15, 20]
Number of operations per job	[5, 7]
Number of machines	[5, 10]
Percentages of alternative machines	[50, 80]
Process time (unit time)	[5, 7]

Table 4. Ratio of the value of Nasr and Elsayed's heuristic solution to the value of the GA solution.

	Mean ratio	Minimum ratio	Maximum ratio
	1.059	0.950	1.184
Using Nasr and Elsayed's solution as a starting solution	1.093	1.004	1.253

In this study, we performed computational experiments with the following values for the genetic algorithm parameters. The parameters are decided after a pilot test.

- Population size: 20.
- Elitist strategy (the best individual was always retained from generation to generation).
- The termination condition was achieved when the number of generations reached 1000.
- Crossover rate: 0.8.
- Mutation rate: 0.01.

In table 4 we compare the ratio of the objective value of Nasr and Elsayed's (1990) heuristic to that of the genetic algorithm. The genetic algorithm outperformed Nasr and Elsayed's (1990) heuristic in 83 out of 100 problems. Obviously, if we use Nasr and Elsayed's (1990) solution as a starting solution for the genetic algorithm, we may obtain better results. The results are also shown in table 4.

5. Conclusions

We have developed an MILP formulation for a job shop system with alternative machine routings. Various objective functions can be used without changing the formulation considerably. The formulations can be used either to compute the optimal solutions for small-sized problems or to test the performance of existing (or new) heuristic algorithms. In addition, a genetic algorithm has been developed for solving large-sized problems. The computational results show that the genetic algorithm outperforms the existing heuristic. Developing other kinds of meta-heuristics and comparing them with the genetic algorithm developed in this paper would be an interesting research problem.

Acknowledgements

The authors are grateful for the comments of the referees. This work was supported by the Korea Research Foundation Grant funded by the Korean Government (MOEHRD) (The Regional Research Universities Program / Research Center for Logistics Information Technology).

References

Baker, K., *Introduction to Sequencing and Scheduling*, 1974 (Wiley: New York).

Baker, K. and Scudder, G., Sequencing with earliness and tardiness penalties: a review. *Oper. Res.*, 1990, **38**, 22–36.

Garey, M. and Johnson, D., *Computers and Intractability: A Guide to the Theory of NP-Completeness*, 1979 (W.H. Freeman: San Francisco).

Gen, M. and Cheng, R., *Genetic Algorithms & Engineering Design*, 1997 (Wiley: New York).

Hutchison, J., Leong, K., Snyder, D. and Ward, P., Scheduling approaches for random job shop flexible manufacturing systems. *Int. J. Prod. Res.*, 1991, **29**, 1053–1067.

Kim, Y., Kim, J. and Lee, W., A multi-level symbiotic evolutionary algorithm for FMS loading problems with various flexibilities. *J. Kor. Inst. Ind. Engrs*, 2003, **29**, 65–77.

Kim, Y., Kim, J. and Shin, K., The integration of FMS process planning and scheduling using an asymmetric multileveled symbiotic evolutionary algorithm. *J. Kor. Inst. Ind. Engrs*, 2004, **30**, 130–145.

Liang, M. and Dutta, S., A mixed-integer-programming approach to the machine loading and process planning problem in a process layout environment. *Int. J. Prod. Res.*, 1990, **28**, 1471–1484.

Liao, C., Optimal control of jobs for production systems. *Comput. Ind. Engng*, 1992, **22**, 163–169.

Moon, I. and Yun, W., The distribution free job control problem. *Comput. Ind. Engng*, 1997, **32**, 109–113.

Nasr, N. and Elsayed, E., Job shop scheduling with alternative machines. *Int. J. Prod. Res.*, 1990, **28**, 1595–1609.

Park, J., Choi, H. and Kim, Y., A genetic algorithm approach to job shop scheduling considering alternative process plans. *J. Kor. Inst. Ind. Engrs*, 1998, **24**, 130–145.

Wilhelm, W. and Shin, H., Effectiveness of alternative operations in a flexible manufacturing system. *Int. J. Prod. Res.*, 1985, **23**, 65–79.

Zhou, H., Feng, Y. and Han, L., The hybrid heuristic genetic algorithm for job shop scheduling. *Comput. Ind. Engng*, 2001, **40**, 191–200.