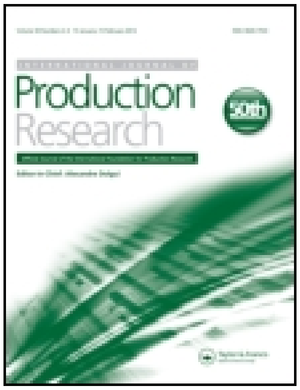
This article was downloaded by: [Seoul National University]
On: 16 February 2014, At: 21:31
Publisher: Taylor & Francis
Informa Ltd Registered in England and Wales Registered Number: 1072954 Registered office: Mortimer House, 37-41 Mortimer Street, London W1T 3JH, UK



International Journal of Production Research
Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/tprs20

Hybrid genetic algorithm for test bed scheduling problems

Anh-Dung Do Ngoc[a], Soo-Heon Lee[b] & Ilkyeong Moon[c]

[a] Department of Mechanical and Manufacturing Engineering, Aalborg University, Aalborg, Denmark.

[b] STX Heavy Industries Co., Ltd., Changwon, Kyungnam, Korea.

[c] Department of Industrial Engineering, Seoul National University, Seoul, Korea.
Published online: 26 Sep 2013.

To cite this article: Anh-Dung Do Ngoc, Soo-Heon Lee & Ilkyeong Moon (2014) Hybrid genetic algorithm for test bed scheduling problems, International Journal of Production Research, 52:4, 1074-1089, DOI: 10.1080/00207543.2013.838327

To link to this article: http://dx.doi.org/10.1080/00207543.2013.838327

PLEASE SCROLL DOWN FOR ARTICLE

Taylor & Francis makes every effort to ensure the accuracy of all the information (the “Content”) contained in the publications on our platform. However, Taylor & Francis, our agents, and our licensors make no representations or warranties whatsoever as to the accuracy, completeness, or suitability for any purpose of the Content. Any opinions and views expressed in this publication are the opinions and views of the authors, and are not the views of or endorsed by Taylor & Francis. The accuracy of the Content should not be relied upon and should be independently verified with primary sources of information. Taylor and Francis shall not be liable for any losses, actions, claims, proceedings, demands, costs, expenses, damages, and other liabilities whatsoever or howsoever caused arising directly or indirectly in connection with, in relation to or arising out of the use of the Content.

This article may be used for research, teaching, and private study purposes. Any substantial or systematic reproduction, redistribution, reselling, loan, sub-licensing, systematic supply, or distribution in any form to anyone is expressly forbidden. Terms & Conditions of access and use can be found at http://www.tandfonline.com/page/terms-and-conditions

International Journal of Production Research, 2014
Vol. 52, No. 4, 1074–1089, http://dx.doi.org/10.1080/00207543.2013.838327

Taylor & Francis
Taylor & Francis Group

Hybrid genetic algorithm for test bed scheduling problems

Anh-Dung Do Ngoc[a], Soo-Heon Lee[b] and Ilkyeong Moon[c]*

[a]*Department of Mechanical and Manufacturing Engineering, Aalborg University, Aalborg, Denmark;* [b]*STX Heavy Industries Co., Ltd., Changwon, Kyungnam, Korea;* [c]*Department of Industrial Engineering, Seoul National University, Seoul, Korea*

(*Received 22 July 2012; accepted 13 August 2013*)

In this paper, we address the scheduling problem for a heavy industry company which provides ship engines for shipbuilding companies. Before being delivered to customers, ship engines are assembled, tested and disassembled on the test beds. Because of limited test bed facilities, it is impossible for the ship engine company to satisfy all customers' orders. Therefore, they must select the orders that can be feasibly scheduled to maximise profit. An integer programming model is developed for order selection and test bed scheduling but it cannot handle large problems in a reasonable amount of time. Consequently, a hybrid genetic algorithm (GA) is suggested to solve the developed model. Several experiments have been carried out to demonstrate the performance of the proposed hybrid GA in scheduling test beds. The results show that the hybrid GA performs with an outstanding run-time and small errors in comparison with the integer programming model.

Keywords: production scheduling; test bed; integer programming; genetic algorithm

1. Introduction

Due to the rapid growth of global trade, the demand for shipbuilding has been continuously increasing resulting in increased demand for ship engine manufacturing. Therefore, we investigate the manufacturing process for ship engines (2-stroke) in a heavy industry company. The engine manufacturing process consists of assembly and testing conducted in a work space called the 'test bed'. The process of securing and utilising the material is removed from our consideration of the manufacturing process because specialised vendors supply these components for high productivity and stable quality of the finished product. The huge engine (over 70 cm in diameter) requires disassembly for shipment after the test run because of the restriction on equipment for transportation and loading at shipyards.

The specifications and delivery of a ship engine manufactured by make-to-order systems are determined in the contract phase of a project, and the delivery of a ship engine must meet a due date to accommodate the overall shipbuilding schedule. Plans for assembly and test runs in the test bed help maximise the efficiency of the equipment and facility used to create and deliver finished engines, which allows customers to stock them for availability when needed. Because the heavy manufacturing company must meet the delivery due date, any necessary timing adjustments must be made in the assembly and test run schedule. If the facilities can accommodate customer orders, then the firm can fulfil them with minimum inventory. However, if the facilities cannot accommodate the order, then the conditions of the contract (e.g. the due date) are inappropriate and must be changed, or the ability to process the order may need to be reconsidered.

A customer order is called a 'project' in the heavy industry company, and in the case presented herein, the studied project involves the assembly, testing and disassembly of a ship engine, for delivery to the shipyard. In this study, we focus on the manufacturing process associated with the test bed: assembly and testing the engine. Specifically, we work on a means to help companies determine the most profitable combinations among potential projects while considering both deadlines and capacity limitations. Because manual scheduling is not the best method to decide on the best projects to pursue in a rapidly changing demand market, we suggest a mathematical model to maximise the total profit by defining all requisites for scheduling within the engine manufacturing process.

The methodologies for solving the scheduling problem are numerous and several studies offer reviews of the selection process for project combinations that maximise total profits. Perez and Wesley (1979) developed the algorithm for find-path/obstacle-avoidance. Utilised extensively in the field in robotics, this algorithm allows for the study of the

---

*Corresponding author. Email: ikmoon@snu.ac.kr path of an object that does not collide into obstacles. O'Rourke et al. (1982) developed an algorithm to search for the intersecting point between two polygons. Lee (1992) regarded the scheduling problem by considering spatial constraints and an axis representing time in a methodology for the spatial allocation of a convex polygon in a rectangle. Morad and Zalzala (1999) studied the integration of process planning and scheduling. They based their formulation on a multi-objective weighted-sum optimization to minimise makespan, total rejects produced and the total cost of production. They proposed a hybrid genetic algorithm (GA) to solve the problem.

Many studies feature the application a hybrid GA to various scheduling problems. Moon, Silver, and Choi (2002) developed a hybrid GA that offered an improvement over the best heuristic, discussed to date in the literature, on the economic lot scheduling problem. Moon, Cha, and Bae (2006) developed a hybrid GA for the problem of group technology economic lot scheduling. Moon et al. (2007) considered a scheduling problem of the shipyard sub-assembly process. They introduced a skid conveyor system in a shipbuilding company and developed a mathematical model and a GA for the shipyard sub-assembly process. Moon, Cha, and Kim (2008) proposed a mixed integer programming model to minimise the maximum storage space requirement over an infinite time horizon by offsetting the inventory cycles of items. They also developed a GA to find the near-optimal solution. Moon, Lee, and Bae (2008) developed a hybrid GA for the scheduling problem, in which alternative routes are allowed in the manufacturing process. Zandieh, Mozaffari, and Gholami (2010) studied the problem of a hybrid flexible flow line where some constraints are considered to alleviate the chasm between the real-world industry scheduling and production scheduling theories. They proposed a GA with an applied response surface methodology to set the parameters of the GA. Abdelhafiez and Alturki (2011) developed a heuristic approach, namely a shaking optimization algorithm, to solve the job shop scheduling problem. Karaoglan and Altiparmak (2011) proposed a hybrid GA based on simulated annealing for the location routing problem with simultaneous pickup and delivery. Gen and Lin (2012) developed a multi-objective hybrid GA for job shop scheduling models.

The purpose of this study is to find the most profitable combination of projects to meet customer demand by a deadline. This scheduling problem has not been studied in the literature because of two folds: (a) this problem is a hybrid problem in which making decisions on selecting profitable projects combined with scheduling selected projects and (b) we have to consider the additional space and time constraints for the test process of huge engines in scheduling. Using data for 2009 and 2010, the investigation details potential solutions for the scheduling problem at the test bed of a heavy industry company. To address the programme with a spatial scheduling solution, a rectangle was used to determine the space and time for engine assembly. The heuristic from the GA was developed to solve the problem such that the best combination of candidate projects is selected for the maximum profit. The integer programme (IP) was analysed with CPLEX (an optimisation software) from ILOG, and the GA and hybrid GA experiments were implemented through a visual basic programme.

The paper is organised as follows. The manufacturing and scheduling method for the ship engine is explained in Section 2, and Section 3 deals with the IP. Section 4 explains the GA and the experiments using the GA. The comparisons of the hybrid GA and the IP are discussed in Section 5. Finally, the conclusions in Section 6 end this paper.

2. Scheduling for the ship engine

2.1 *Assembly, testing and disassembly of a ship engine*

In this study, we focus on the manufacturing processes of the test bed, namely those of assembly, testing and disassembly. After assembled on a test bed, engines are tested. After that, testing process is executed to check engines assembled in assembly operation. Finally, the huge engines are disassembled for delivery to customers. The assembly process is conducted simultaneously with the main assembly, piping and electrical work. The bed plate of an engine being assembled is placed on several fixed rails of the test bed. Other parts of the engine are also assembled on the bed plate. A bigger engine requires more fixed test bed rails than does a small one. The test process includes a takeover check, installation, domestication, pretest and official trial. The takeover check and installation allow for verification that all parts are in order and without defect. A dynamometer is loaded for testing the output of the engine. In addition, the piping for fuel and water, cabling, and valve conditions are checked. An engine with the diameter over 70 cm is considered a huge engine and additional fixed rails are required for testing with the dynamometer. However, additional fixed rails are not needed for testing smaller engines.

2.2 *Test bed scheduling by Microsoft Excel*

The engine is assembled and tested in a test bed, which can be modelled in the Excel computer programme. Five test beds were used in the heavy industry in which a case study was undertaken. Each bed has fixed rails into which the

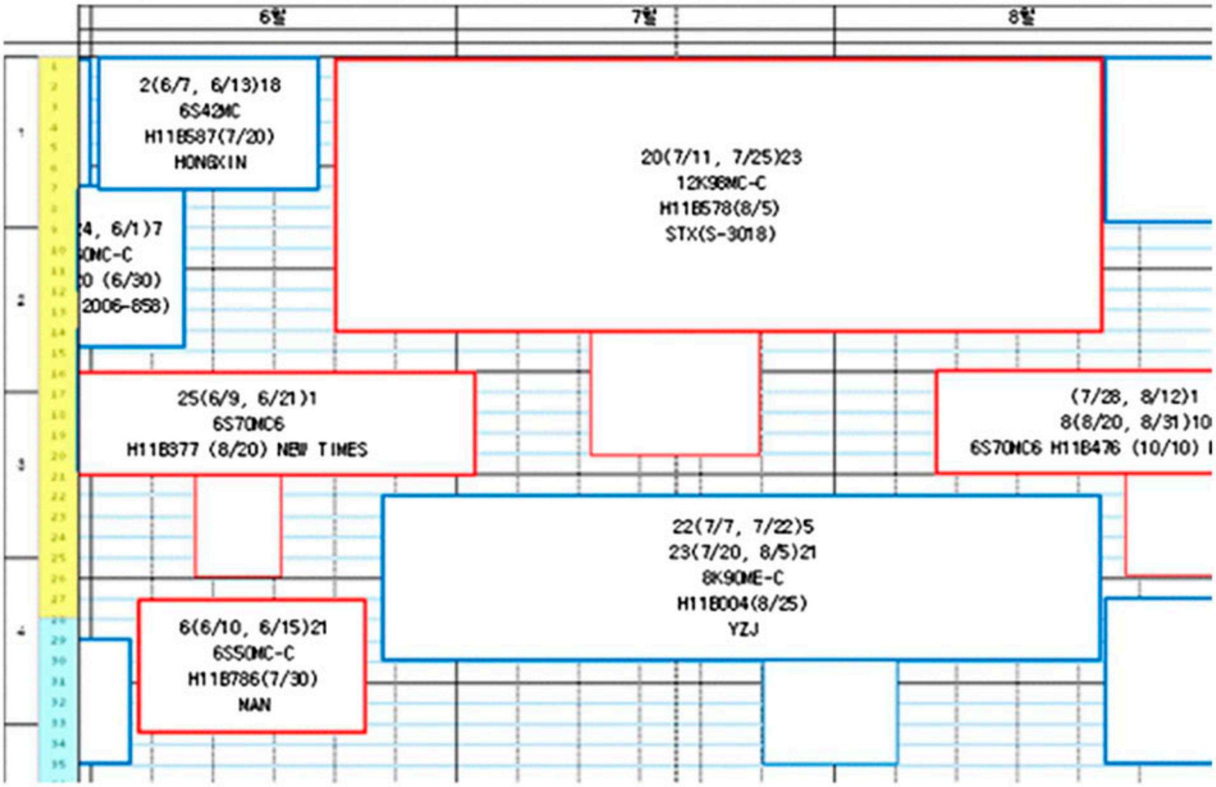


Figure 1. Result of test bed scheduling (Excel).

engine and dynamometer are pegged at 2 m intervals. In Figure 1, a test bed is represented by a cell that contains the values 1, 2, 3 or 4 in the first column.

The manufacturing process of the ship engine is represented as a rectangle in which the length, measured as number of columns, is the duration of the project, and the width, measured by rows, is the number of units of a resource. In this study, one Excel column represents one period, and one Excel row represents one fixed rail. Therefore, the length of the rectangle is the number of periods required to manufacture a ship engine, and the width of the rectangle shows the number of fixed rails on the test bed occupied during the time of engine manufacturing. The result of test bed scheduling, as manually computed, is shown as the allocation of the rectangles on an Excel sheet.

The schedule for a huge engine is comprised of two rectangles, as shown in Figure 2. The large rectangle represents the manufacturing process of the corresponding engine while the small rectangle represents the number of periods and number of fixed rails occupied during the test run for that huge engine. For the small engine, only one rectangle is used to represent the assembly and the test run. Small engines take less time to make because the disassembly process is unnecessary.

The contents in the rectangle display the dates of each process, type of engine, project number, shipyard and hull number of the ship. In Figure 2, 20 in the first line represents the date that the bed plate of this engine is allocated in

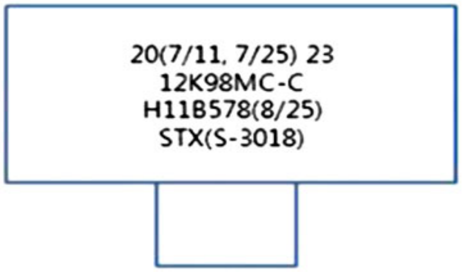


Figure 2. Representation of a huge engine on the test bed schedule.

Table 1. Data for the test bed scheduling example.

Project	Due date	Processing time (days)			Huge engine?
		Assembly	Testing	Disassembly	
1	8	4	1	2	Yes
2	7	4	1	–	
3	10	3	1	2	Yes
4	14	4	1	3	Yes
5	8	3	1	–	
6	6	3	1	–	

the test bed (i.e. the assembly start date is 20 June), and 7/11 and 7/25 in parentheses represent the start and finish dates for the test run: 11 July and 25 July, respectively. In the first line, 23 represents the finish date of disassembly after an official trial: 23 August. In the second line, 12K98MC-C indicates the type of engine, and H11B578 in the next line stands for the project number. The right side of the project number shows the delivery date: 25 August. The last line displays the shipyard and hull number of the ship.

The procedure to schedule the manufacturing of ship engines in test beds is described as follows. First, the huge engine projects are sorted in ascending order by due date. Next, the schedule of each huge engine is assigned if space is available. Second, the small engine projects are sorted in ascending order by due date. Then, each small engine is assigned a space, if possible. To assign an order by start date of manufacturing is a fundamental principle of scheduling. To illustrate the above procedure, we consider an example with 6 project candidates: suppose 10 fixed rails are available in a planning horizon of 14 days. The due dates as well as assembly, testing and disassembly times of each project are given in Table 1. Each huge engine needs three fixed rails for assembly and disassembly and one fixed rail for testing. Each small engine requires two fixed rails for assembly and no fixed rails for testing. Applying the above procedure, Projects 1, 3, 4, 5 and 6 are selected, and the schedule is shown in Figure 3. Project 2 cannot be chosen because no space is available for it.

3. Mathematical model

The mathematical model helps with the decision to select and schedule a combination of ship engine projects to maximise profit in a planning horizon. In a production scheduling problem, the objective function is usually to minimise makespan. However, in this case study, the company receives many orders (or 'projects') but possesses limited test bed facilities. The company cannot satisfy all orders and so must deny some. A good schedule allows the company to satisfy as many orders as possible by selecting the best combinations of orders and scheduling them. The mathematical model is presented in terms of an IP based on the following assumptions:

- The project, or order, is for the manufacture of a ship engine.
- The project schedule depends on the manufacturing processes of the ship engine.

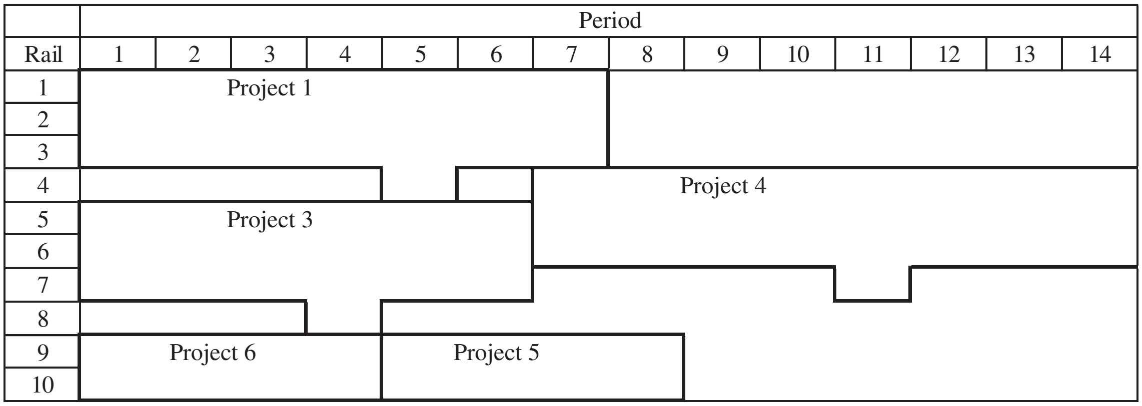


Figure 3. Assembly schedule for the test bed scheduling example.

- A project must be finished by or before the due date.
- Assembly, testing and disassembly times of a ship engine are pre-determined.
- The manufacturing process of a ship engine cannot be pre-empted.
- The manufacturing process of ship engines must be scheduled within the planning horizon.

Indices

i	index for a candidate project, $i \in \{1, 2, 3, \ldots, I\}$, in which I is the number of candidate projects
t	a planning period, $t \in \{1, 2, 3, \ldots, T\}$, in which T is the last period of the planning horizon
n	fixed rail where a ship engine is assembled, $n \in \{1, 2, 3, \ldots, N\}$, in which N is the number of fixed rails in the test bed
N_i	a set of fixed rails appropriate for assembling the engine according to project i
N_h	a set of fixed rails used to assemble a huge engine

Parameters

p_i	processing time of project i
a_i	assembly time of project i
b_i	testing time of project i
c_i	cost of project i
s_i	profit of project i
d_i	due date for project i
k_i	number of fixed rails required to assemble and disassemble project i
m_i	number of fixed rails required to test project i

Decision variables

$X_i =$	1, if project i is selected 0, otherwise
$Y_{int} =$	1, if the assembly of project i is started in period t at fixed rail n (where n is the first rail) 0, otherwise
$Z_{int} =$	1, if fixed rail n is occupied to process project i in period t 0, otherwise

Mathematical model

$$\text{Max} \sum_{i=1}^{I} (s_i - c_i) X_i$$

subject to

$$\sum_{n \in N_i} \sum_{t=1,\ t \leqslant T}^{d_i - p_i} Y_{int} = X_i \quad \forall i \tag{1}$$

$$Z_{in't'} \geq Y_{int} \quad \forall i,\ n \in N_i, t, n \leq n' \leq n + k_i - 1,\ t \leq t' \leq t + p_i - 1 \tag{2a}$$

$$Z_{in't'} \geq Y_{int} \quad \forall i,\ n \in N_i,\ t,\ n + k_i \leq n' \leq n + k_i + m_i - 1,\ t + a_i \leq t' \leq t + a_i + b_i - 1 \tag{2b}$$

$$\sum_{i=1}^{I} Z_{int} \leq 1 \quad \forall n,\ t \tag{3}$$

$$\sum_{i=1}^{I}\sum_{n\in N_h}\sum_{\substack{t'\geq t-(a_i+b_i-1)\\ t\,\leq\, t-a_i}}^{T} Y_{i,n-k_i,t}\leq 1\quad\forall t \tag{4}$$

$$X_i, Y_{\text{int}}, Z_{\text{int}}\in\{0,1\}\quad\forall i,\ n,\ t$$

- The objective function is to maximise total revenue. Constraint (1) implies that if project i is selected, then it should be finished by or before its due date. It is illustrated in Figure 4. The two rectangles represent the manufacturing processes of a huge engine.

If the assembly of project i is started at fixed rail n in period t, then several adjacent rails need to be used during the processing time of project i. This is represented by Constraints (2a) and (2b) in Figure 5.

Constraint (3) stipulates that in a certain period, a fixed rail is used to assemble one, and only one, ship engine for a project. See Figure 6 for a violation of Constraint (3).

In a certain period, only one huge engine can be tested at the fixed rails that are used to assemble it. This is represented by Constraint (4). See Figure 7.

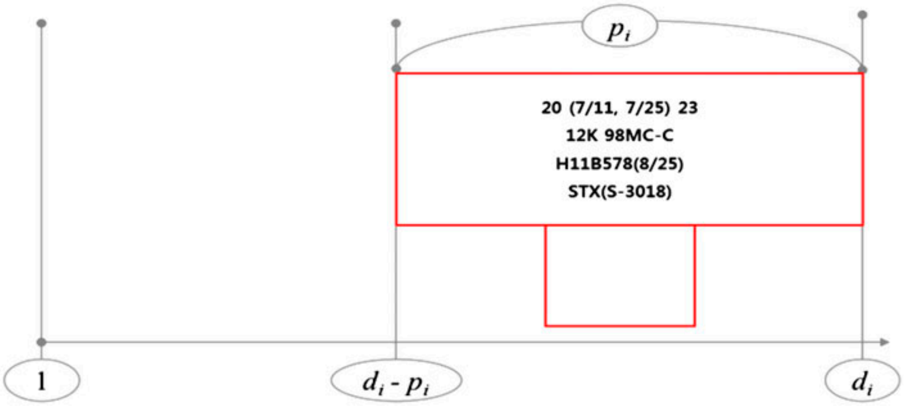


Figure 4. Description of Constraint (1).

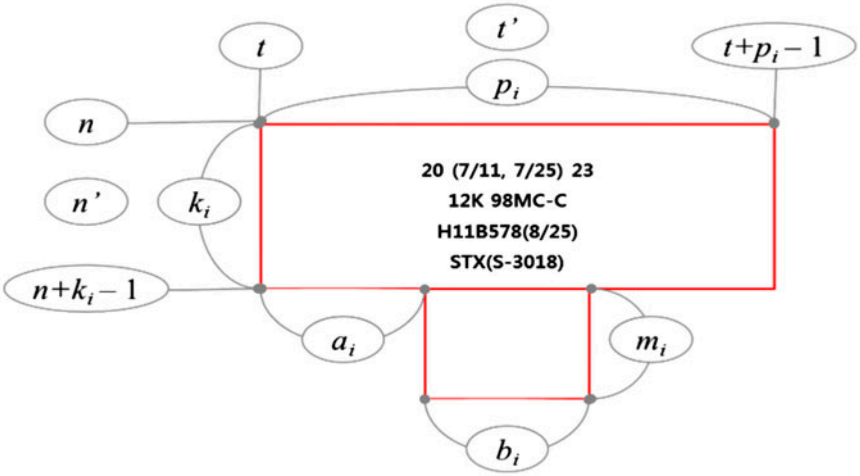


Figure 5. Description of Constraints (2a) and (2b).

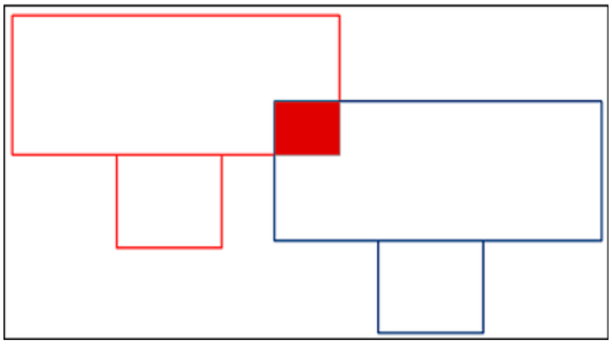


Figure 6. Description of Constraint (3).

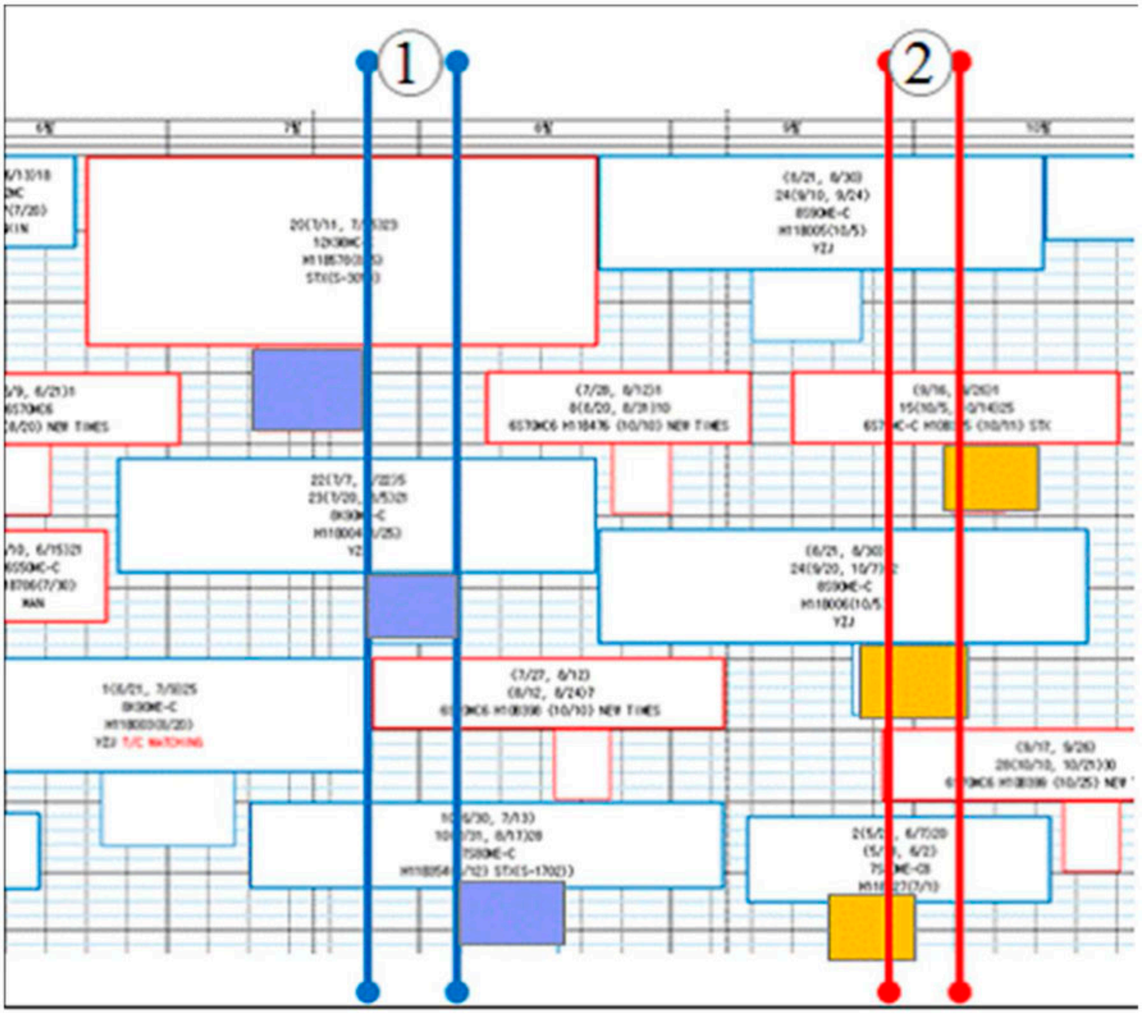


Figure 7. Description of Constraint (4).

4. Genetic algorithms

Large-scale decision-making for ship engine manufacturing is very difficult. In a heavy industry company, the decision-making process is too complicated to ensure maximise revenue. Although solutions to small problem can be attained using optimization software, most problems require too much computation time to make the method feasible. However, the GA can typically be used to solve large problems (Kachitvichyanukul 2012).

GAs have been shown to be highly efficient for solving this type of scheduling problem and therefore were applied. The GA is a meta-heuristic method used to obtain a near-optimal solution using computer programming based on the principles of evolution and natural selection (Gen and Cheng 2000). This heuristic is powerful as well as time- and cost-effective, and it initializes the population by the string in integers or real numbers to express the gene of the biological structure (Goldberg 1989). GAs allow for flexibility because they impose no requirement for formulating solutions in a particular way, and the objective function(s) can be differentiable, continuous, linear or separable; it need not reflect any particular data type. Thus, GAs can be applied to any problem (e.g. single or multi-objective, single or multi-level, linear or nonlinear) that can be encoded and the quality of the solution computed. GAs can be easily combined with exact solution algorithms (e.g. branch and bound), local search (LS) (i.e. memetic algorithm), and/or other (meta-) heuristics and guarantee local optimality of the solution or improve convergence patterns (Golias et al. 2010). The GA is expected to obtain the optimal solution through the random search process associated with 'selection', 'crossover', and 'mutation' operations of regenerated chromosomes.

4.1 *Chromosome coding, decoding, and fitness evaluation*

In this study, a chromosome is a solution structured as a string of binary numbers. A gene represents a candidate project. The value of a gene is 0 or 1. If the value of the gene is 1, the corresponding project is selected. Figure 8 illustrates the chromosome in an example with nine candidate projects. Projects 1, 4, 6, 7 and 9 are selected, and the other projects are not selected.

The solution is obtained from the chromosome through a simple heuristic algorithm. First, the chromosome is decoded to obtain solution for the variable X_i. If the value of gene i is 1, then the project i is selected and $X_i = 1$; otherwise, project i is not selected and $X_i = 0$. A list of selected projects is determined after all genes are considered. Second, these selected projects are ordered by using the earliest due date rule. If two or more selected projects have the same due date, then they are sorted in the descending order of revenue (i.e. profit–cost). Finally, the selected projects are scheduled by their order. If project i cannot be scheduled, the value of X_i is changed to zero meaning that project i is not selected. In this heuristic algorithm, the project is scheduled so that it can be started as early as possible. The values of Y_{int} and Z_{int} are determined throughout the procedure to schedule the selected project. The objective function value is calculated from the solution that is obtained by the algorithm and also represents the fitness of the corresponding chromosome.

Fitness of chromosome i = the objective function value calculated from the solution represented by chromosome i =

$$\text{Max} \sum_{k=1}^{K} (s_k - c_k) X_k$$

where X_k is obtained after decoding chromosome i.

4.2 *GA parameters and operators*

4.2.1 *GA parameters*

After pilot runs, the GA parameters were set as follows: The population size is 40, and the maximum number of generations is 100. Eighty offspring are created in each iteration, the crossover probability is 0.9, and the mutation rate is 0.08.

4.2.2 *Selection*

Many types of selection methods could be used: roulette wheel, tournament, and ranking. After conducting a pilot study, we applied the tournament selection in this study.

4.2.3 *Crossover*

The crossover operator is used to create new offspring. Types of crossover include one-point, two-point, uniform, partial matched and order. In our investigation, based on the results of a pilot study, we applied the uniform crossover.

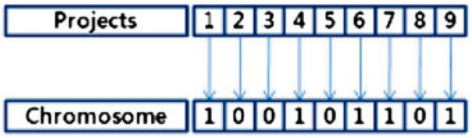


Figure 8. Structure of a chromosome.

4.2.4 *Mutation*

The mutation is used to maintain the diversity of genes thorough the restoration of lost characteristics from ancestors. Among the many types of mutation, we used the flip-bit mutation for our study such that the value associated with the selected gene will be changed from 0 to 1 and vice versa.

4.3 ***Numerical experiments of the GA***

We conducted the experiments with a small problem to evaluate the GA. Data on the four candidate projects are shown in Table 2.

At first, we selected four projects with a due date of day 11. For the small engines (Projects 1 and 4), we needed no additional rails for the test run (b_i), and we assumed that with a test time of zero the assembly time (a_i) equals the processing time (p_i). However, for the huge engines (Projects 2 and 3), more rails are required for testing than are used for assembly. Therefore, we distinguished the testing time from assembly time. For Project 2, an additional rail is required for the testing process, and the processing time is the total of the assembly, testing and disassembly times. See Figure 9.

The fixed rails allocated to the engine are numbered between 1 and 5, and the dynamometer used for the testing process is located between rails 1 and 6. We conducted the experiment with test data by CPLEX and obtained the results shown in Table 3. Figure 10 shows the schedule of the selected projects.

We conducted the second test with a different condition that delays the due date by one day. In this case, we expected that all projects will be selected. The project data are shown in Table 4, and the results of the experiment are shown in Table 5 and Figure 11.

Table 2. Data for the first test.

Project	p_i	a_i	b_i	c_i	s_i	$s_i - c_i$	d_i	k_i	m_i	N_i	N_h
1	4	4	0	2	5	3	11	2	0	1–4	1–6
2	5	3	1	3	7	4	11	3	1	1–3	1–6
3	6	4	1	4	9	5	11	3	1	1–3	1–6
4	3	3	0	1	3	2	11	2	0	1–4	1–6

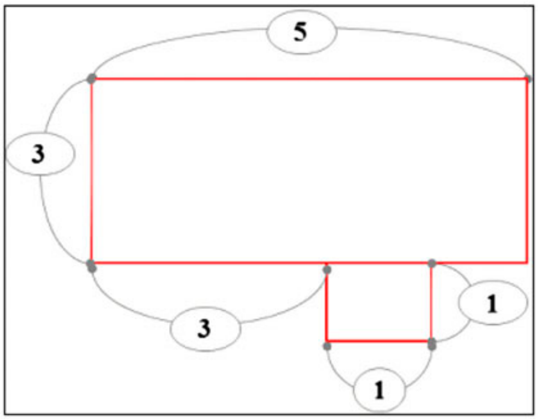


Figure 9. Data for Project 2 in the small-size problem.

Table 3. Results of the first test by CPLEX.

Project	Selected	Start rail	Start period
1	Y	4	2
2	N	–	–
3	Y	1	4
4	Y	2	1

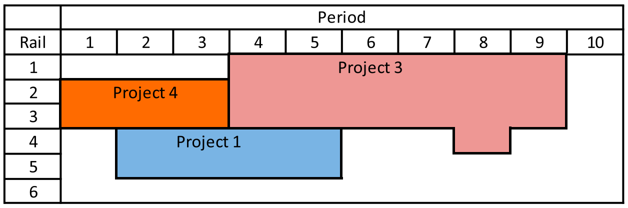


Figure 10. Production schedule for the first test.

Table 4. Data for the second test.

Project	p_i	a_i	b_i	c_i	s_i	$s_i - c_i$	d_i	k_i	m_i	N_i	N_h
1	4	4	0	2	5	3	12	2	0	1–4	1–6
2	5	3	1	3	7	4	12	3	1	1–3	1–6
3	6	4	1	4	9	5	12	3	1	1–3	1–6
4	3	3	0	1	3	2	12	2	0	1–4	1–6

Table 5. Results of the second test by CPLEX.

Project	Selected	Start rail	Start period
1	Y	4	1
2	Y	1	7
3	Y	1	1
4	Y	4	7

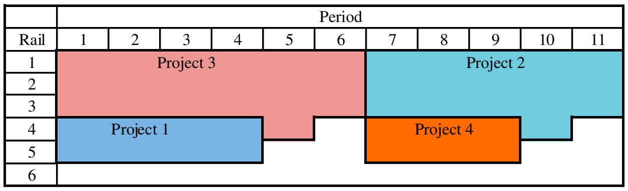


Figure 11. Production schedule for the second test.

In the third test, the number of fixed rails is increased by one rail (see Table 6 for all project data). We expected that all projects would be selected. The results are shown in Table 7 and Figure 12.

For the fourth test, we shifted to a sooner due date while keeping the number of fixed rails as for the third test. The data for the fourth test are obtained by changing the due date of Project 1 to the fifth day, the due date for Project 3 to the seventh day and the due date for Project 4 to the fifth day. Table 8 shows the data for the fourth test. The results are presented in Table 9 and Figure 13.

The results in Tables 3, 5, 7 and 9 were obtained by using CPLEX, a well-known optimisation software, to solve the testing problem. In addition, we solved the same problem by using the GA. The results of the GA using the same data as for the first four tests are shown in Tables 10–13 and Figures 14–17.

Based on the results, one can see that the GA can find the optimal solution in the first three test problems. However, when the due dates are very tight, as in the last problem, the pure GA cannot find the optimal solution. We, therefore,

Table 6. Data for the third test.

Project	p_i	a_i	b_i	c_i	s_i	$s_i - c_i$	d_i	k_i	m_i	N_i	N_h
1	4	4	0	2	5	3	11	2	0	1–5	1–7
2	5	3	1	3	7	4	11	3	1	1–4	1–7
3	6	4	1	4	9	5	11	3	1	1–4	1–7
4	3	3	0	1	3	2	11	2	0	1–5	1–7

Table 7. Results of the third test by CPLEX.

Project	Selected	Start rail	Start period
1	Y	5	7
2	Y	4	2
3	Y	1	4
4	Y	1	1

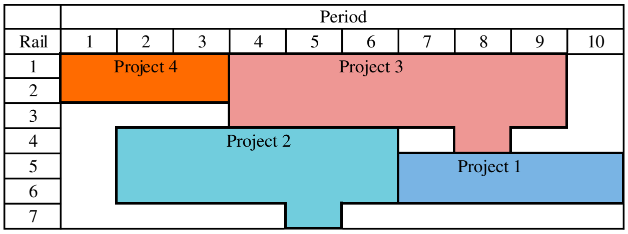


Figure 12. Production schedule for the third test.

Table 8. Data for the fourth test.

Project	p_i	a_i	b_i	c_i	s_i	$s_i - c_i$	d_i	k_i	m_i	N_i	N_h
1	4	4	0	2	5	3	5	2	0	1–5	1–7
2	5	3	1	3	7	4	11	3	1	1–4	1–7
3	6	4	1	4	9	5	7	3	1	1–4	1–7
4	3	3	0	1	3	2	5	2	0	1–5	1–7

Table 9. Results of the fourth test by CPLEX.

Project	Selected	Start rail	Start period
1	Y	4	1
2	Y	4	6
3	Y	1	1
4	Y	6	2

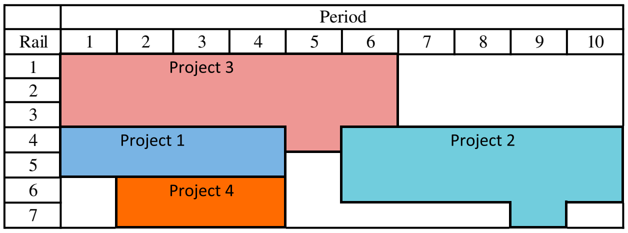


Figure 13. Production schedule for the fourth test.

developed a hybrid GA to improve the quality of the solution. In this study, we combine the pure GA with a LS algorithm to improve the fitness of the chromosome, which in turn improves the solution.

5. Hybrid GA

For the four tests, the IP can obtain an optimal solution, but large problems take long computation times. The pure GA can solve problems quickly, but the quality of the solution in the fourth test was poor. Therefore, we introduce a hybrid GA to obtain better solutions.

Table 10. Results of the first test by GA.

Project	Selected	Start rail	Start period
1	Y	4	1
2	N	–	–
3	Y	1	1
4	Y	1	7

Table 11. Results of the second test by GA.

Project	Selected	Start rail	Start period
1	Y	4	1
2	Y	1	7
3	Y	1	1
4	Y	5	5

Table 12. Results of the third test by GA.

Project	Selected	Start rail	Start period
1	Y	4	1
2	Y	4	6
3	Y	1	1
4	Y	6	1

Table 13. Results of the fourth test by GA.

Project	Selected	Start rail	Start period
1	Y	1	1
2	Y	3	4
3	N	–	–
4	Y	3	1

5.1 *Procedure*

The LS applied to non-selected projects creates the difference between pure and hybrid GAs. After finding a feasible schedule for the selected projects, the LS is applied. Non-selected projects are listed in descending order of revenue. After that, each non-selected project is considered. If a considered project can be inserted into the schedule, the value of the corresponding gene will be changed from 0 to 1. The following LS procedure breaks down the process by step:

Step 1 Sort the non-selected projects in descending order of revenue.
Step 2 Consider the project at the top of the list.
If the project can be assigned to the schedule, then
- change the corresponding gene to 1 and
- update the fitness of the chromosome.

Step 3 Remove the project from the list.
Step 4 If there is no project in the list, then stop. Otherwise, go to Step 2.

The structure of the hybrid GA is shown in Figure 18.

5.2 *Numerical experiment of the hybrid GA*

The results of the pure GA differ from those of the IP only in the fourth test. The purpose of the hybrid GA is to improve the results of the pure GA. Therefore, to test the hybrid GA, we used data from the fourth test of the small

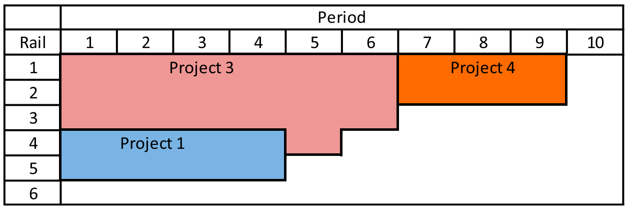


Figure 14. Production schedule for the first test by GA.

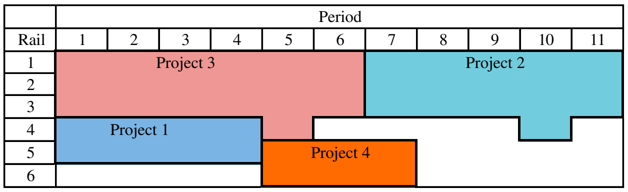


Figure 15. Production schedule for the second test by GA.

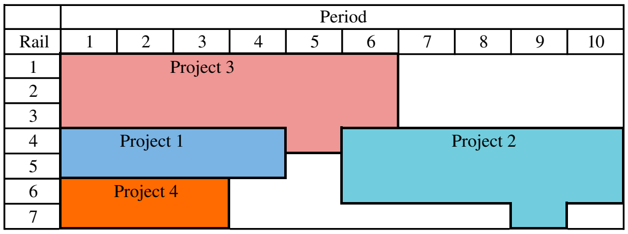


Figure 16. Production schedule for the third test by GA.

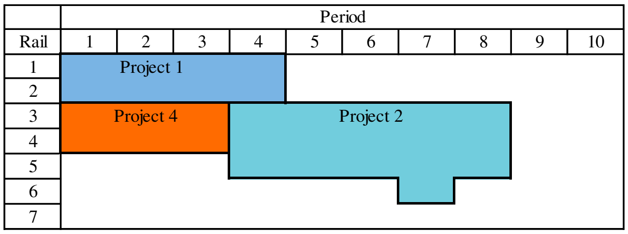


Figure 17. Production schedule of the fourth test by GA.

problem as illustrated in Table 8 and re-conducted only the fourth experiment. The results are shown in Figure 19, and one can see the improved solution created by the hybrid GA.

To further evaluate the performance of the hybrid GA, we conducted the numerical experiments with data from the first quarter of 2009 as reported by a heavy industry company. These data are shown in Table 14. The result of the hybrid GA is compared to that of the IP.

Using the real-world data, the result of the IP is obtained by using CPLEX from ILOG, as shown in Table 15. Projects 1, 2 and 3 are not selected because they cannot be processed in a reasonably timely manner. The value of the objective function was 4.13, and the computation time was 102 s. The results of the hybrid GA are shown in Table 16.

The result of the hybrid GA and the IP are the same, but the computation time of the hybrid GA is very short: 19 s. This finding suggests that the hybrid GA can handle a large problem with a near-optimal solution. The experiments for each quarter from 2009 to 2010 are shown in Table 17. The upper bounds shown in the third column are obtained by using CPLEX. If the upper bound equals the best objective value found by the IP, then the CPLEX found the optimal solution (in Table 17, the problem for the first quarter of 2009).

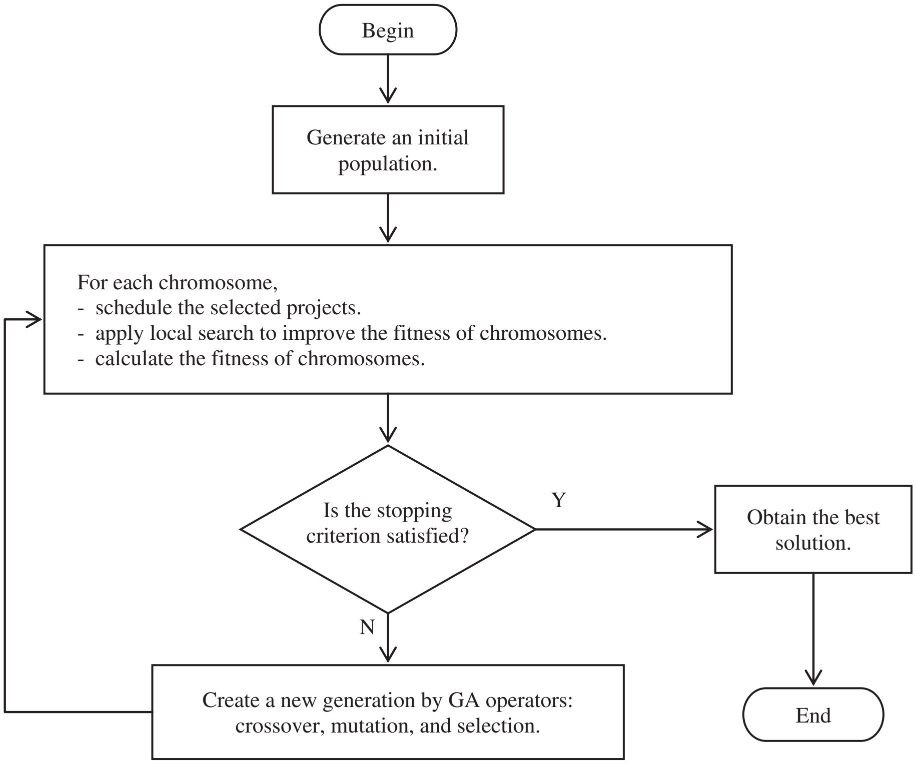


Figure 18. Structure of the hybrid GA.

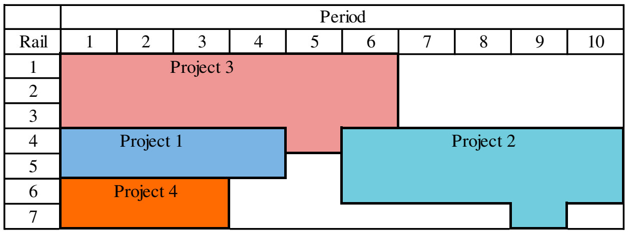


Figure 19. Production schedule for the fourth test as obtained through the hybrid GA.

Table 14. Data for the first quarter of 2009.

Project	p_i	a_i	b_i	c_i	s_i	d_i	k_i	m_i	N_i	N_h
1	22	22	0	1.59	2.14	15	4	0	1–70	–
2	22	22	0	2.96	3.07	20	4	0	1–70	–
3	22	22	0	2.76	3.05	20	4	0	1–70	–
4	23	23	0	3.19	3.68	53	5	0	1–69	–
5	23	23	0	3.65	4.19	68	5	0	1–69	–
6	22	22	0	1.7	2.14	69	4	0	1–70	–
7	30	14	5	4.7	5.58	70	6	5	1–46	1–52
8	22	22	0	2	2.41	76	4	0	1–70	–
9	22	22	0	3.16	4.53	84	4	0	1–70	–

Table 15. Results of the IP for the first quarter of 2009.

Project	Selected	Start rail	Start period
1	N	–	–
2	N	–	–
3	N	–	–
4	Y	1	1
5	Y	25	11
6	Y	30	16
7	Y	6	1
8	Y	17	1
9	Y	21	8

Table 16. Results of the hybrid GA for the first quarter of 2009.

Project	Selected	Start rail	Start period
1	N	–	–
2	N	–	–
3	N	–	–
4	Y	69	7
5	Y	69	30
6	Y	63	35
7	Y	1	1
8	Y	15	37
9	Y	70	61

Table 17. Comparing the results between the IP and the hybrid GA by quarter (Q).

			Best found objective value		Computation time (in minutes)		
Period	Number of projects	Upper bound	IP	HGA	IP	HGA	% Error (IP/HGA)
2009 Q1	9	4.13	4.13	4.13	1.7	0.3	0
2009 Q2	24	19.75	15.52	19.75	>1020	2.4	0
2009 Q3	27	20.81	17.37	20.81	>780	2.3	0
2009 Q4	39	–	–	26.60	–	10.1	–
2010 Q1	26	15.49	12.56	15.24	>750	3	1.61
2010 Q2	30	15.85	15.67	15.85	>40	16.8	0
2010 Q3	30	14.79	12.28	14.79	>390	2.9	0
2010 Q4	40	–	–	26.46	–	15.6	–

The IP requires a longer time to obtain a solution than does the hybrid GA. The difference between the two methods is shown in the last column of Table 17. In the case that the IP cannot obtain an optimal solution due to insufficient computer memory, the difference can be calculated using the objective value found by the hybrid GA and the upper bound provided by CPLEX. The upper bounds of the fourth quarter of 2009 and 2010 in the IP are not provided because of the limitations of software capability.

6. Conclusions

The purpose of this study is to find a combination of the most profitable projects that can satisfy the capacity constraints for the test bed scheduling problem. We conducted the experiments by the IP and a hybrid GA with data of a heavy industry company and compared the results. The IP obtains an optimal solution but takes too much time for large problems. The pure GA can solve quickly, but the quality of the results is not good enough to use for scheduling. The hybrid GA is proposed to improve the solution of the pure GA. We conducted several experiments to evaluate the hybrid GA. The results show that the hybrid GA has an outstanding run-time despite small errors. The hybrid GA can obtain a near-optimal solution. We expect that the results of this study will be applied not only to manufacturing scheduling but also will become the standard for strategic decision making to maximise profit and thus ensure global competitiveness. However, we recognise that some factors are not considered in this study, such as the initial status of fixed rails and labour capacity. These factors will be investigated in further research we will conduct to obtain more realistic results.

The findings from this study will be used as a standard for making choices on the combination of projects that maximises profitability. A company that chooses to offer a set of product lines and items, in all combinations, in response to variable customer demands will benefit from a means of determining profitability. Ultimately, the combination of projects that maximises profitability should be determined before a contract with a customer is signed.

Acknowledgements

The authors are grateful for the useful comments from two anonymous referees.

Funding

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MEST) (No. 2012R1A2A2A01012355).

References

Abdelhafiez, E. A., and F. A. Alturki. 2011. “A Shaking Optimization Algorithm for Solving Job Shop Scheduling Problem.” *Industrial Engineering and Management Systems* 10: 7–14.

Gen, M., and R. Cheng. 2000. *Genetic Algorithms and Engineering Optimization*. New York: Wiley.

Gen, M., and L. Lin. 2012. “Multiobjective Genetic Algorithm for Scheduling Problems in Manufacturing.” *Industrial Engineering and Management Systems* 11: 310–330.

Goldberg, D. 1989. *Genetic Algorithms in Search, Optimization and Machine Learning*. Reading, MA: Addison-Wesley.

Golias, M., M. Boilé, S. Theofanis, and C. Efstathiou. 2010. “The Berth Scheduling Problem: Maximizing Berth Productivity and Minimizing Fuel Consumption and Emissions Production.” *Transportation Research Record: Journal of the Transportation Research Board, Marine Transportation and Port Operations* 2166: 20–27.

Kachitvichyanukul, V. 2012. “Comparison of Three Evolutionary Algorithms: GA, PSO, and DE.” *Industrial Engineering and Management Systems* 11: 215–223.

Karaoglan, I., and F. Altiparmak. 2011. “A Hybrid Genetic Algorithm for the Location-routing Problem with Simultaneous Pickup and Delivery.” *Industrial Engineering and Management Systems* 10: 24–33.

Lee, K. J. 1992. “Spatial Scheduling and its Application to Ship Building.” Master’s thesis, Department of Management Science, Korea Advanced Institute of Science and Technology.

Moon, I. K., H. C. Bae, K. C. Park, and B. C. Cha. 2007. “Scheduling of Shipyard Sub-assembly Process Using Genetic Algorithms.” *IE Interfaces* 20: 34–40.

Moon, I. K., B. C. Cha, and H. C. Bae. 2006. “Hybrid Genetic Algorithm for Group Technology Economic Lot Scheduling Problem.” *International Journal of Production Research* 44: 4551–4568.

Moon, I. K., B. C. Cha, and S. K. Kim. 2008. “Offsetting Inventory Cycles Using Mixed Integer Programming and Genetic Algorithm.” *International Journal of Industrial Engineering* 15: 245–256.

Moon, I. K., S. H. Lee, and H. R. Bae. 2008. “Genetic Algorithm for Job Shop Scheduling Problem with Alternative Routings.” *International Journal of Production Research* 46: 2695–2705.

Moon, I. K., E. A. Silver, and S. Choi. 2002. “Hybrid Genetic Algorithm for the Economic Lot-scheduling Problem.” *International Journal of Production Research* 40: 809–824.

Morad, N., and A. Zalzala. 1999. “Genetic Algorithms in Integrated Process Planning and Scheduling.” *Journal of Intelligent Manufacturing* 10: 169–179.

O’Rourke, J., C. Chien, T. Olson, and D. Naddor. 1982. “A New Linear Algorithm for Intersecting Convex Polygons.” *Computer Graphics and Image Processing* 19: 384–391.

Perez, T. L., and M. A. Wesley. 1979. “An Algorithm for Planning Collision-free Paths among Polyhedral Obstacles.” *Communications of ACM* 22: 560–570.

Zandieh, M., E. Mozaffari, and M. Gholami. 2010. “A Robust Genetic Algorithm for Scheduling Realistic Hybrid Flexible Flow Line Problems.” *Journal of Intelligent Manufacturing* 21: 731–743.