

An improved genetic algorithm for multidimensional optimization of precedence-constrained production planning and scheduling

Son Duy Dao¹ · Kazem Abhary¹ · Romeo Marian¹

Received: 21 April 2016 / Accepted: 20 December 2016 / Published online: 9 January 2017
© The Author(s) 2017. This article is published with open access at Springerlink.com

Abstract Integration of production planning and scheduling is a class of problems commonly found in manufacturing industry. This class of problems associated with precedence constraint has been previously modeled and optimized by the authors, in which, it requires a multidimensional optimization at the same time: what to make, how many to make, where to make and the order to make. It is a combinatorial, NP-hard problem, for which no polynomial time algorithm is known to produce an optimal result on a random graph. In this paper, the further development of Genetic Algorithm (GA) for this integrated optimization is presented. Because of the dynamic nature of the problem, the size of its solution is variable. To deal with this variability and find an optimal solution to the problem, GA with new features in chromosome encoding, crossover, mutation, selection as well as algorithm structure is developed herein. With the proposed structure, the proposed GA is able to “learn” from its experience. Robustness of the proposed GA is demonstrated by a complex numerical example in which performance of the proposed GA is compared with those of three commercial optimization solvers.

Keywords Genetic algorithm · Optimization · Precedence constraint · Integration of planning and scheduling · Variable-length chromosome

Introduction

Integration of production planning and scheduling is a class of problems commonly found in manufacturing industry. One of the important constraints in the integration of production planning and scheduling is a so called precedence constraint. This constraint has two classes, namely hard precedence constraint and soft precedence constraint. Hard precedence constraint is a constraint that makes the solution infeasible if violated; while, soft precedence constraint imposes a penalty only and the solution is still feasible. In this article, the production planning and scheduling problems, associated with both hard and soft precedence constraints, are considered.

The precedence-constrained production planning and scheduling is a multidimensional optimization problem, in which a number of sub-problems such as production selection, product allocation, manufacturing sequence, etc. are required to be simultaneously solved. From computational complexity theory point of view, the precedence-constrained production planning and scheduling is a NP-hard problem. It should be noted that NP is a technical term in computational complexity theory in computer science and mathematics, which stands for Non-deterministic Polynomial-time. NP problems are the set of decision problems that can be solved by non-deterministic polynomial-time bounded Turing machines (Cadoli et al. 2000). In addition, NP-hard is the class of decision problems which are as hard as any NP problem (Shapiro and Delgado-Eckert 2012). NP-hard problems are algorithmically solvable but computationally intractable (Shapiro and Delgado-Eckert 2012). There is no exact method that can find the global optimal solutions to NP-hard problems in polynomial time, and fast approximate heuristics and meta-heuristics are the popular approaches to search for high-quality/practical solutions (He et al. 2012).

✉ Son Duy Dao
son.dao@mymail.unisa.edu.au

¹ School of Engineering, University of South Australia, Mawson Lakes Campus, Mawson Lakes, SA 5095, Australia

In this article, an improved Genetic Algorithm (GA) is developed to search for optimal solution to the precedence-constrained production planning and scheduling problem.

Literature review

As mentioned before, there are a number of sub-problems in the precedence-constrained production planning and scheduling problem. The sub-problems are inter-connected to each other. In other words, the precedence-constrained production planning and scheduling is an integrated optimization problem. There are three approaches to solve this problem, namely hierarchical, iterative, and full-space approaches (Maravelias and Sung 2009). In the first two approaches, the integrated problem is decomposed into a master sub-problem and a slave sub-problem. The master sub-problem is used to determine production planning while the slave sub-problem is for scheduling. The slave sub-problem uses the output of the master sub-problem as its input. In the full-space approach, the planning and scheduling are fully integrated and they are simultaneously solved. There is no doubt that the hierarchical and iterative approaches can solve large-scale problems because the search space of the problem is significantly reduced, due to the decomposition; however, they have a very limit chance to find global optimal solution. In contrast, as solving the planning and scheduling simultaneously, the full-space approach has the highest potential to obtain global optimal solution.

Meta-heuristics are popular optimization algorithms, often used to solve large-scale complex optimization problems in various fields (Abtahi and Bijari 2016; Javanmard and Koraeizadeh 2016; Moradgholi et al. 2016). In the research of Shao et al. (2009), a genetic algorithm-based method was used for optimization of two functions: process planning and scheduling. However, these two processes are not fully integrated. Therefore, the global optimal solution to the problem could not be found. In addition, an artificial intelligent search algorithm, named symbiotic evolutionary algorithm, was developed by Kim et al. (2003) for the integration of process planning and job shop scheduling. Again, the way of constructing the entire solutions proposed in that algorithm prevents the entire solution from global optimization. Furthermore, Liu and Fang (2012) proposed a heuristic based approach to deal with the integration, in which the entire problem is divided into a number of sub-problems and sub-constraint based interval planning algorithm is developed for each sub-problem. Nevertheless, the interaction between the planning and scheduling is very limited. Therefore, it is very hard to obtain the global optimal solution to the integrated problem.

Although, there have been a large number of methods developed to deal with the integration of production planning and scheduling, methods based on full-space approach are still very limited, especially for the integration of precedence-constrained production planning and scheduling problem. In the previous research works (Dao and Marian 2011a, b), to deal with the complex constraints, two modified GAs with variable chromosome sizes were developed to solve this problem in one production line environment. Later on, the precedence-constrained production planning and scheduling problem, associated with multiple production line environment, were solved by Dao and Marian (2011c, 2013) with more advanced GAs. Nevertheless, the global optimal solution to this problem has not been obtained yet since the solution method is not fully integrated.

To overcome this limitation, a more robust GA is developed herein to improve the quality of the solutions to the precedence-constrained production planning and scheduling problem with multiple production lines. With a new chromosome encoding, modified crossover/mutation/selection operations as well as a modified algorithm structure, the developed GA is capable of searching for the global optimal solution to the problem, with very high success rate.

Problem statement

As mentioned before, the main purpose of this article is to further develop a more robust GA for the precedence-constrained production planning and scheduling problem with multiple production lines. It is noted that the problem under consideration was already described in the previous publications (Dao and Marian 2011c, 2013) and it is restated herein, as follows, for the readers' convenience.

There is a manufacturing company which produces a number of different products with a number of different production lines. Manufacturing resources of the company such as labor, material and working capital are limited. Currently, there are a number of product orders with a variety of products and delivery deadlines, to be fulfilled. Manufacturing cost, fixed cost, labor, and product changeover in different production lines are different. The company is capable of producing any mix of types of products and it plans to produce at least D different types of products in the next period of time. In addition, penalty cost due to late delivery and returned product are applied. Question now is (1) determine the types of products to be produced (planning), (2) determine the number of the products in each selected type to be made (planning), (3) determine the allocation of the selected products to production lines (scheduling), and (4) determine the sequence to produce the selected products in each production line (scheduling); so that the company will have maximized



profit and customer satisfaction index while all given constraints are simultaneously satisfied.

Model formulation

The mixed integer formulation for this problem is developed as follows.

Assumptions

- The company can produce any mix of different products.
- The company can work 24 h a day, 7 days a week.
- The proceedings from selling products will only be available for next period of time, so they should be ignored for the current planning horizon-current period of time.

Indices

-
- i = Production line index
 - j = Manufacturing sequence index
 - k = Product index
-

Parameters

-
- α = Weight coefficient
 - A = Number of points of customer satisfaction index lost each day of delay when each product is made after its deadline
 - B = Number of points of customer satisfaction index obtained when each product is made before its deadline
 - C = Total working capital available (\$)
 - D_k = Deadline of product k (day of month)
 - E_{ki} = Manufacturing expense of product k in the production line i (\$)
 - M = Total material available (kg)
 - MR_k = Material requirement for product k (kg)
 - l = Number of different production lines
 - L = Total labor available (hours)
 - LR_{ki} = Labour requirement for product k in the production line i (hours)
 - O_i = Other overhead for running production line i (\$/hour)
 - p = Number of different product types available
 - PC_i = Product changeover in the production line i (hours)
 - U_k = Penalty due to late delivery of product k (percentage of the product price per day of delay)
 - V = Minimum number of product varieties required
 - R_k = Maximum number of delay days accepted with penalty for product k ; otherwise the customer does not accept the product and it will be returned to the company
 - S_k = Selling price of product k (\$)
-

Decision variables

-
- C_i = Working capital allocated to production line i (\$)
 - M_i = Material allocated to production line i (kg)
 - L_i = Labor allocated to production line i (hours)
 - P_{kij} = Product k selected among the available ones and made in production line i in sequence of j
 - Q_k = Quantity of product k
 - Q_{ki} = Quantity of product k made in production line i
 - $H_{kij} = \begin{cases} 1 & \text{If product } k \text{ is selected to be made in} \\ & \text{production line } i \text{ in sequence } j \\ 0 & \text{Otherwise} \end{cases}$
 - $\theta_i = \begin{cases} 1 & \text{If products in sequence } j \text{ and in sequence } j - 1 \\ & \text{in production line } i \text{ are not the same} \\ 0 & \text{Otherwise} \end{cases}$
 - $\text{Sign1}\{x\} = \begin{cases} x & \text{If } x > 0 \\ 0 & \text{If } x \leq 0 \end{cases}$
 - $\text{Sign2}\{x\} = \begin{cases} 1 & \text{If } x > 0 \\ 0 & \text{If } x \leq 0 \end{cases}$
 - Unique (x) = A function which is able to determine the number of unique elements in the matrix x
-

Objective function

Multi-objective function is used to take into account both company profit and customer satisfaction index. The objective function to be maximized is sum of the total profit of the company and its customer satisfaction index with given weight coefficient, which is calculated by Eq. (1) where: F is fitness value; TI is total income; MC is total manufacturing cost; OH is total overhead for running the selected production lines; CD is total cost associated with penalty due to products made after deadline; CR is total cost due to returned products or not accepted by customer because they are too late; SI is total points of customer satisfaction index; α is weight coefficient.

$$F = \alpha[\text{TI} - \text{MC} - \text{OH} - \text{CD} - \text{CR}] + (1 - \alpha) \text{SI}. \tag{1}$$

The cost components in Eq. (1) are computed as follows:

The total income that the company earns from the product orders (TI):

$$\text{TI} = \sum_{k=1}^p Q_k \cdot S_k \tag{2}$$

The total manufacturing cost of the product orders (MC):

$$\text{MC} = \sum_{k=1}^p \sum_{i=1}^l Q_{ki} \cdot E_{ki} \tag{3}$$

The total overhead for running the selected production lines (OH):

$$OH = \sum_{k=1}^p \sum_{i=1}^l \sum_{j=1}^{\infty} (\theta_i \cdot PC_i + LR_{ki}) H_{kij} O_i \quad (4)$$

The total cost associated with penalty due to products made after deadline (CD):

$$CD = \sum_{k=1}^p \sum_{i=1}^l \sum_{j=1}^{\infty} \left[\frac{\text{Sign1}\{(\theta_i \cdot PC_i + LR_{ki}) H_{kij} - 24D_k\}}{24} \right] U_k \cdot S_k \quad (5)$$

The total cost due to returned products or not accepted by customer because they are too late (CR):

$$CR = \sum_{k=1}^p \sum_{i=1}^l \sum_{j=1}^{\infty} [\text{Sign2}\{(\theta_i \cdot PC_i + LR_{ki}) H_{kij} - 24D_k - 24R_k\}] S_k \quad (6)$$

The total points of customer satisfaction index (SI):

$$SI = \sum_{k=1}^p \sum_{i=1}^l \sum_{j=1}^{\infty} \left[B - \left(\frac{\text{Sign1}\{(\theta_i \cdot PC_i + LR_{ki}) H_{kij} - 24D_k\}}{24} \right) A \right] \quad (7)$$

Subject to:

Quantity of product k :

$$\sum_{i=1}^l Q_{ki} = Q_k \quad (8)$$

The total working capital available:

$$\sum_{i=1}^l C_i = C \quad (9)$$

The total material available:

$$\sum_{i=1}^l M_i = M \quad (10)$$

The total labour available:

$$\sum_{i=1}^l L_i = L \quad (11)$$

The total material allocated to production line i :

$$\sum_{k=1}^p Q_{ki} MR_k \leq M_i \quad (12)$$

The total labour allocated to production line i :

$$\sum_{k=1}^p \sum_{j=1}^{\infty} (\theta_i \cdot PC_i + LR_{ki}) H_{kij} \leq L_i \quad (13)$$

The total working capital allocated to production line i :

$$\sum_{k=1}^p Q_{ki} \cdot E_{ki} + \sum_{k=1}^p \sum_{j=1}^{\infty} (\theta_i \cdot PC_i + LR_{ki}) H_{kij} O_i \leq C_i \quad (14)$$

Minimum number of product varieties required:

$$p \geq \text{Unique} \left(\sum_{k=1}^p \sum_{i=1}^l \sum_{j=1}^{\infty} P_{kij} \right) - 1 \geq V. \quad (15)$$

Optimization methodology

A modified GA with new features in chromosome encoding, mutation, crossover, selection operation as well as GA structure is proposed herein to optimize the multidimensional integration problem of precedence-constrained production planning and scheduling in multiple production line environment as described above. This approach is fully integrated implying that it leaves no boundary between the planning and scheduling. Therefore, the proposed approach can enhance the chance of achieving the globally optimal solution for the problem.

In addition, a new GA structure modified from the traditional GA is proposed herein. With the proposed structure, the GA can “learn” from its experience. Because GA is a search heuristic that mimics the process of natural evolution, it cannot guarantee to find the best solution after only one run (Marian et al. 2008a, b). Therefore, the proposed GA is designed to run a number of times. Each run has a certain number of generations, defined as a civilization in this paper. Moreover, from the second civilization, the GA does not start searching from the beginning. The proposed GA is designed to be able to “remember” some information, several good or best chromosomes for examples, from the previous civilization. As a result, the initial population of the proposed GA, except the first civilization, is not totally generated at random. A certain number of chromosomes in the initial population are randomly generated as usual and the rest is transferred, modified or repaired, if necessary, from the previous civilization. That is why the proposed GA has the ability to “learn” from its experience.

The proposed structure of the GA might cause a problem, i.e., premature convergence to local optimum because the fitness values of the good chromosomes from the previous civilization are much superior to those generated at random. As a result, those good chromosomes are constantly selected for the next generations if the conventional selection operator is used. To resolve this issue, a principle is proposed herein according to which every chromosome can be selected only once in one generation. After a chromosome is selected for the next generation, it is removed from the current pool of the candidates.

The major components of the proposed GA are presented as follows:

Chromosome encoding

Each production line can be encoded as a string corresponding to products allocated to each production line and their manufacturing sequences. Accordingly, each chromosome encoding a solution for the problem consists of *N* strings corresponding to the *N* production lines. It should be noted that the length of each string may be different from each other because of the resource constraints. Each chromosome has two parts: resource allocation and manufacturing sequence as illustrated in the Table 1. Without loss of generality and to make it convenient, the problem with three production lines is considered for the rest of the paper.

The chromosome in Table 1 represents a sample solution for the problem with three different production lines and 50 different types of products. The first part of the chromosome is the resource allocation as highlighted in the yellow cells. The L column is the allocation of the company’s labour to three production lines. And the M and C columns are the allocations of the material and working capital to three production lines, respectively; and the values in the yellow cells are in percentage. That is why summation of each column is always equal to 100%.

The second part of the chromosome is the manufacturing sequence as highlighted in the green cells. The values in these cells represent the corresponding product types and the locations of the cells represent the corresponding manufacturing sequences. In the example, it is assumed that there are 50 different types of products representing by 50 numbers from 1 to 50. Therefore, the values in the green cells can be any number from 0 to 50 where “0” means that no product is allocated to the corresponding location. As the labour, material, and working capital allocated to production lines are different, the length of each production line can be different from each other as shown in the Table 1. Moreover, even if the available resources allocated to the production lines are exactly the same, the lengths of the production lines could be different. That is because the types of products and the producing sequences of the products in each production line are not the same.

It can be clearly seen that it is not easy to randomly generate a feasible chromosome as described above because it involves a number of complex constraints. In order to generate a feasible chromosome, the following steps are proposed.

- Step 1 Randomly generate the resource part of the chromosome.
- Step 2 Determine the labour, material, and working capital allocated to *i*th production line. Let them be L_i , M_i , and C_i , respectively.
- Step 3 Generate the *j*th product to be made in the production line *i* by selecting one product at random and adding it to the production line as sequential order. Let the current production schedule in this production line be $PL_{i(1 \rightarrow j)}$, where $(1 \rightarrow j)$ is a series of numbers: 1, 2, 3, ... *j*, representing the sequential order of the selected products made in *i*th production line.
- Step 4 Calculate the cumulative summations of labour (CL_i), material (CM_i), and working capital (CC_i) required in $PL_{i(1 \rightarrow j)}$.
- Step 5 Check:
If $CL_i \leq L_i$ and $CM_i \leq M_i$ and $CC_i \leq C_i$ then $j = j + 1$ and Go to Step 3
Else Go to Step 6.
- Step 6 Determine the feasible schedule in *i*th production line which is $PL_{i(1 \rightarrow j-1)}$.
- Step 7 Calculate $i = i + 1$ and Go to Step 2. Steps 2–7 are to be repeated until all of the production lines are generated.
- Step 8 Check if the number of different types of products is greater than the minimum requirement *V* as represented by Eq. (15) then Stop. Otherwise repeat the Steps 1–8.

Evaluation

Quality of the solutions to the problem (quality of the chromosomes) is determined based on the summation of the total profit and the customer satisfaction index with a weight coefficient as shown in Eq. (1). Chromosomes with better qualities have more chance to be passed to the next generations. Selection operation of the proposed GA will be presented in “Selection operation”.

Table 1 A typical chromosome with three production lines

	Resource Allocation			Manufacturing Sequence																							
	L	M	C	1	2	3	4	5	6	7	8	9	8	6	12	33	45	1	23	22	49	49	32	2	12	0	0
Production line 1	20	60	18	2	4	34	50	2	7	8	9	8	6	12	33	45	1	23	22	49	49	32	2	12	0	0	0
Production line 2	45	15	12	4	2	4	10	3	3	3	9	33	9	44	26	3	37	48	50	29	40	45	4	5	23	22	3
Production line 3	35	25	70	19	11	6	36	7	9	8	43	23	2	50	47	2	26	27	18	20	0	0	0	0	0	0	0

Crossover operation

In principle, crossover is a simple cut and swap operation. Due to the nature of constraint and chromosome, a modified crossover operation is required. In this study, crossover operation applying to both resource allocation and manufacturing sequence parts of a chromosome is proposed.

Crossover 1—crossover operation applying to resource allocation part

The labour column, material column, and working capital column in the resource allocation part of a chromosome are constrained such that the summation of numbers in each column is equal to 100. Therefore, the principle of the Crossover 1, which is the swap of two different groups of the columns in two different chromosomes, is proposed herein. As complex constraints involved, the offspring chromosomes must be repaired after the swap operation to make sure that they are feasible. The following steps are proposed for the Crossover 1.

Step 1 Randomly select two parent chromosomes as shown in two sub-tables in Table 2.

- Step 2 Randomly select one cutting point in the resource allocation part, the red lines highlighted in Table 2, for example.
- Step 3 Swap the two groups of the columns as shown in the resource allocation parts of the chromosomes in Table 3.
- Step 4 Determine the new allocations of the labour, material, and working capital to each production line in the two offspring chromosomes.
- Step 5 + Trim the products at the end of each production line schedule if the current resources required are exceeded the new allocated resources determined in Step 4.
+ Add new random products at the end of each production line schedule while satisfied the new allocated resources determined in Step 4.
+ Otherwise, leave the offspring chromosomes in Step 3 as they are.
- Step 6 Check the number of different types of products selected in each chromosome in Step 5. If it satisfies the requirement in Eq. (15) then Stop; Otherwise, repeat Steps 1–6.

It should be noted that the repair strategy used in Step 5 is that the length of each production line is adjusted based

Table 2 Two parent chromosomes and one cutting point

	Resource Allocation			Manufacturing Sequence																							
	L	M	C	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
Production line 1	20	60	18	2	4	34	50	2	7	8	9	8	6	12	33	45	1	23	22	49	49	32	2	12	0	0	0
Production line 2	45	15	12	4	2	4	10	3	3	3	9	33	9	44	26	3	37	48	50	29	40	45	4	5	23	22	3
Production line 3	35	25	70	19	11	6	36	7	9	8	43	23	2	50	47	2	26	27	18	20	0	0	0	0	0	0	0

	Resource Allocation			Manufacturing Sequence																							
	L	M	C	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
Production line 1	15	65	30	3	43	2	1	9	6	11	23	43	22	33	50	0	0	0	0	0	0	0	0	0	0	0	0
Production line 2	25	20	30	50	3	9	43	45	50	2	3	4	6	4	16	32	37	23	42	21	29	30	3	9	0	0	0
Production line 3	60	15	40	39	23	9	10	17	19	9	20	32	8	49	45	5	30	23	43	21	28	39	45	0	0	0	0

Table 3 Two offspring chromosomes after crossover 1

	Resource Allocation			Manufacturing Sequence																							
	L	M	C	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
Production line 1	20	65	30	2	4	34	50	2	7	8	9	8	6	12	33	45	1	23	22	0	0	0	0	0	0	0	0
Production line 2	45	20	30	4	2	4	10	3	3	3	9	33	9	44	26	3	37	48	50	29	40	45	0	0	0	0	0
Production line 3	35	15	40	19	11	6	36	7	9	8	43	23	2	50	47	2	26	27	18	20	34	23	23	43	2	3	0

	Resource Allocation			Manufacturing Sequence																							
	L	M	C	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
Production line 1	15	60	18	3	43	2	1	9	6	11	23	43	22	33	50	45	32	21	49	50	46	47	47	23	32	34	2
Production line 2	25	15	12	50	3	9	43	45	50	2	3	4	6	4	16	32	37	0	0	0	0	0	0	0	0	0	0
Production line 3	60	25	70	39	23	9	10	17	19	9	20	32	8	49	45	5	30	23	43	21	28	0	0	0	0	0	0

on the available resources. If available resources are not enough, the products at the end of a production line will be removed until the production line meets the resource limitations. If available resources are enough, the production line will be kept the same. If available resources are redundant, some products will be generated randomly and added at the end of the production line until almost all available resources have been utilized. It is also noted that it is quite hard to use all of the available resources due to the discrete nature of the problem. Therefore, utilization of almost all available resources is acceptable. Feasible offspring chromosomes, which satisfy all of the given constraints, would look like as shown in Table 3.

As the allocations of the resources are changed, the lengths of the production lines in the offspring chromosomes can be different from those of their parent chromosomes as illustrated in Tables 2 and 3.

Crossover 2—crossover operation applying to manufacturing sequence part

As the lengths of different production lines are not the same, the cutting point should be in a certain region. In Table 4, the efficient region in the first and the second chromosomes are from the 1st to 21st product and from the 1st to 15th product, respectively. To make the crossover operation more effective, the region for Crossover 2 is

proposed to be from 1st to 15th product as shown in Table 5. Therefore, the cutting point should be somewhere in the highlighted regions in Table 5, the 8th product for example. After the swap operation, the offspring chromosomes must be repaired to make sure that they are feasible. The following steps are proposed for Crossover 2.

- Step 1 Randomly select two parent chromosomes
- Step 2 Determine the efficient regions in the selected chromosomes as shown in Table 4
- Step 3 Determine the region for Crossover 2 as shown in Table 5
- Step 4 Randomly select one cutting point in the region for Crossover 2 as shown in Table 5
- Step 5 Swap the two parts as shown in Table 6
- Step 6 Determine the resources required in each production line
- Step 7 Repair the manufacturing sequence parts of the offspring chromosomes to make them feasible
- Step 8 Check the number of different types of products selected in each chromosome in Step 7. If the requirement is satisfied then Stop; Otherwise, repeat Steps 1–8.

It is noted that, the repair strategy presented in “Crossover 1—crossover operation applying to resource allocation part” is used again here in Step 7 to adjust the lengths of the production lines to make the offspring

Table 4 Two parent chromosomes and their efficient regions for crossover 2

	Resource Allocation			Manufacturing Sequence																							
	L	M	C	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
Production line 1	20	60	18	2	4	34	50	2	7	8	9	8	6	12	33	45	1	23	22	49	49	32	2	12	0	0	0
Production line 2	45	15	12	4	2	4	10	3	3	3	9	33	9	44	26	3	37	48	50	29	40	45	4	5	23	22	3
Production line 3	35	25	70	19	11	6	36	7	9	8	43	23	2	50	47	2	26	27	18	20	0	0	0	0	0	0	0

	Resource Allocation			Manufacturing Sequence																							
	L	M	C	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
Production line 1	15	65	30	3	43	2	1	9	6	11	23	43	22	33	50	0	0	0	0	0	0	0	0	0	0	0	0
Production line 2	25	20	30	50	3	9	43	45	50	2	3	4	6	4	16	32	37	23	42	21	29	30	3	0	0	0	0
Production line 3	60	15	40	39	23	9	10	17	19	9	20	32	8	49	45	5	30	23	0	0	0	0	0	0	0	0	0

Table 5 Region for crossover 2 and the cutting point for crossover 2

	Resource Allocation			Manufacturing Sequence																							
	L	M	C	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
Production line 1	20	60	18	2	4	34	50	2	7	8	9	8	6	12	33	45	1	23	22	49	49	32	2	12	0	0	0
Production line 2	45	15	12	4	2	4	10	3	3	3	9	33	9	44	26	3	37	48	50	29	40	45	4	5	23	22	3
Production line 3	35	25	70	19	11	6	36	7	9	8	43	23	2	50	47	2	26	27	18	20	0	0	0	0	0	0	0

	Resource Allocation			Manufacturing Sequence																							
	L	M	C	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
Production line 1	15	65	30	3	43	2	1	9	6	11	23	43	22	33	50	0	0	0	0	0	0	0	0	0	0	0	0
Production line 2	25	20	30	50	3	9	43	45	50	2	3	4	6	4	16	32	37	23	42	21	29	30	3	0	0	0	0
Production line 3	60	15	40	39	23	9	10	17	19	9	20	32	8	49	45	5	30	23	0	0	0	0	0	0	0	0	0

feasible. The feasible offspring chromosomes would look like as shown in Table 6. Due to the changes in the production lines, their lengths are changed accordingly as illustrated in Tables 4 and 6.

Mutation operation

Once again, due to the nature of the chromosome, the modified mutation operation is required. Two types of mutations: mutation 1 and mutation 2 which apply to resource allocation part and manufacturing sequence part of a chromosome, respectively, are proposed herein.

Mutation 1–mutation operation applying to resource allocation part

As mentioned in “Crossover 1–crossover operation applying to resource allocation part”, the summation of each column in the resource allocation part must equal 100. Therefore, the Mutation 1 proposed herein is the swapping

two random elements only in one random column in one chromosome. As a result, the constraint above is always satisfied. Similar to the Crossover 1, the offspring chromosome must be repaired after the swap operation to make sure it is feasible. The following steps are proposed for the Mutation 1.

- Step 1 Randomly select one chromosome
- Step 2 Randomly select one column in the resource allocation part of the selected chromosome, column M for example as shown in Table 7
- Step 3 Randomly select two elements in the selected column as shown in Table 7
- Step 4 Swap the two selected elements as shown in Table 8
- Step 5 Determine the new resource allocation in each production line
- Step 6 Repair the manufacturing part of the offspring chromosome based on the new resource allocations

Table 6 Two offspring chromosomes after crossover 2

	Resource Allocation			Manufacturing Sequence																							
	L	M	C	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
Production line 1	20	60	18	2	4	34	50	2	7	8	23	43	22	33	50	45	2	8	9	34	28	25	0	0	0	0	0
Production line 2	45	15	12	4	2	4	10	3	3	3	3	4	6	4	16	32	37	23	42	21	0	0	0	0	0	0	0
Production line 3	35	25	70	19	11	6	36	7	9	8	20	32	8	49	45	5	30	23	24	49	12	1	23	3	0	0	0

	Resource Allocation			Manufacturing Sequence																							
	L	M	C	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
Production line 1	15	65	30	3	43	2	1	9	6	11	9	8	6	12	33	45	1	23	22	0	0	0	0	0	0	0	0
Production line 2	25	20	30	50	3	9	43	45	50	2	9	33	9	44	26	3	37	48	50	29	40	45	4	5	0	0	0
Production line 3	60	15	40	39	23	9	10	17	19	9	43	23	2	50	47	2	26	27	18	20	11	2	3	6	32	23	10

Table 7 Parent chromosome and two random genes for mutation 1

	Resource Allocation			Manufacturing Sequence																							
	L	M	C	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
Production line 1	20	60	18	2	4	34	50	2	7	8	9	8	6	12	33	45	1	23	22	49	49	32	2	12	0	0	0
Production line 2	45	15	12	4	2	4	10	3	3	3	9	33	9	44	26	3	37	48	50	29	40	45	4	5	23	22	3
Production line 3	35	25	70	19	11	6	36	7	9	8	43	23	2	50	47	2	26	27	18	20	0	0	0	0	0	0	0

Table 8 Offspring chromosome after mutation 1

	Resource Allocation			Manufacturing Sequence																							
	L	M	C	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
Production line 1	20	25	18	2	4	34	50	2	7	8	9	8	6	12	33	45	1	23	0	0	0	0	0	0	0	0	0
Production line 2	45	15	12	4	2	4	10	3	3	3	9	33	9	44	26	3	37	48	50	29	40	45	4	5	23	22	3
Production line 3	35	60	70	19	11	6	36	7	9	8	43	23	2	50	47	2	26	27	18	20	4	30	35	41	0	0	0

Step 7 Check the number of different types of products selected in the chromosome in Step 6. If the requirement is satisfied then Stop; Otherwise, repeat Steps 1–7.

It is noted that once again the repair strategy presented in “Crossover 1–crossover operation applying to resource allocation part” is used in Step 6 to adjust the lengths of the production lines to make the chromosome feasible. The feasible offspring chromosome would look like as shown in Table 8. Due to the changes in the resource allocations, the lengths of the production lines are changed as illustrated in Tables 7 and 8.

Mutation 2–mutation operation applying to manufacturing sequence part

As the length of every production line is different, the modified mutation operation is required. The principle proposed herein is that all of the genes selected for Mutation 2 must be in one production line and in somewhere between the first to the last product, so called value region as highlighted in Table 9. It is noted that the value regions in different production lines could be different as

shown in Table 9. The following steps are proposed for implementation of Mutation 2.

- Step 1 Randomly select one parent chromosome as shown in Table 9
- Step 2 Randomly select one production line in the selected chromosome, e.g., the production line 1
- Step 3 Randomly select two genes in the selected production line within the value region as shown in Table 10
- Step 4 Swap the two selected genes as shown in Table 11
- Step 5 Determine the resources required in the selected production line
- Step 6 Repair the manufacturing sequence part of the offspring chromosome based on the available resources
- Step 7 Check the number of different types of products selected in the chromosome in Step 6. If the requirement is satisfied then Stop; Otherwise, repeat Steps 1–7.

It is noted that, the repair strategy presented in “Crossover 1–crossover operation applying to resource allocation part” is also used in Step 6 to adjust the lengths of the production lines

Table 9 A parent chromosome for mutation 2

	Resource Allocation			Manufacturing Sequence																							
	L	M	C	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
Production line 1	20	60	18	2	4	34	50	2	7	8	9	8	6	12	33	45	1	23	22	49	49	32	2	12	0	0	0
Production line 2	45	15	12	4	2	4	10	3	3	3	9	33	9	44	26	3	37	48	50	29	40	45	4	5	23	22	3
Production line 3	35	25	70	19	11	6	36	7	9	8	43	23	2	50	47	2	26	27	18	20	0	0	0	0	0	0	0

Table 10 Parent chromosome and two randomly selected genes for mutation 2

	Resource Allocation			Manufacturing Sequence																							
	L	M	C	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
Production line 1	20	60	18	2	4	34	50	2	7	8	9	8	6	12	33	45	1	23	22	49	49	32	2	12	0	0	0
Production line 2	45	15	12	4	2	4	10	3	3	3	9	33	9	44	26	3	37	48	50	29	40	45	4	5	23	22	3
Production line 3	35	25	70	19	11	6	36	7	9	8	43	23	2	50	47	2	26	27	18	20	0	0	0	0	0	0	0

Table 11 Offspring chromosome after mutation 2

	Resource Allocation			Manufacturing Sequence																							
	L	M	C	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
Production line 1	20	60	18	2	4	34	23	2	7	8	9	8	6	12	33	45	1	50	22	49	49	32	0	0	0	0	0
Production line 2	45	15	12	4	2	4	10	3	3	3	9	33	9	44	26	3	37	48	50	29	40	45	4	5	23	22	3
Production line 3	35	25	70	19	11	6	36	7	9	8	43	23	2	50	47	2	26	27	18	20	0	0	0	0	0	0	0

to make the chromosome feasible. The feasible offspring chromosome would look like as shown in Table 11. Due to the changes in the production line, the length of the production line is changed as illustrated in Tables 9 and 11.

Selection operation

As mentioned before, the proposed GA has the novel structure which can facilitate the “learning ability”. With this structure, some good chromosomes from the previous civilization are recorded and used as the chromosomes in the first generation of the current civilization. As a result, those chromosomes overwhelm the others which might cause the premature convergence to local optimum. To avoid this problem, the modified Roulette Wheel selection method is proposed via the following steps.

- Step 1 Select the best chromosome in the population pool for the next generation
- Step 2 Delete the selected chromosome in Step 1 from the population pool
- Step 3 Determine the summation of fitness values in the updated population pool
- Step 4 Determine the selection probability of every chromosome in the updated population pool
- Step 5 Make a wheel according to these probabilities as illustrated in Fig. 1a
- Step 6 Spin the roulette wheel once and select one chromosome for the next generation
- Step 7 Delete the selected chromosome in Step 6 from the updated population pool
- Step 8 Update the population pool and Go to Step 3
- Step 9 Repeat Steps 3–8 until the number of the selected chromosome equals the population size

Figure 1 illustrates the changes in selection probabilities of the chromosomes when the size of population pool is changed. In Fig. 1a, there are 5 chromosomes to be chosen for the next generation: A, B, C, D, and E. It can be seen that the chromosome C is the best one with the selection probability of 39%. It is a good idea to select the best one first to

make sure it passes to the next generation as it has the highest potential to make a difference. After chromosome C has been selected and removed from the population pool, there are 4 chromosomes to be chosen from and their selection probabilities are as shown in Fig. 1b. Now, this roulette wheel is spun and one chromosome, say chromosome B, is selected for the next generation. Figure 1c presents the selection probabilities of 3 chromosomes after chromosome B is selected and removed from the population pool. As a result, the selection probability of chromosome A, for example, is increased from 6 to 11 or 14%. Clearly, the selection probabilities of the chromosomes are changing and the effect of super chromosomes can be eased as each chromosome can be selected only once, no matter how good it is.

The proposed selection approach has two advantages. The first one is that it can prevent super chromosomes from dominating population by maintaining too many copies in the population which causes the premature convergence to local optimum. The second one is that it is able to maintain the diversity of population in probability fashion so that the population pool can contain much more valuable information for genetic search.

Structure of the proposed genetic algorithm

To improve the search capability, the structure of the proposed GA is developed as shown in Fig. 2. It should be noted that, in the Fig. 2, g is the number of generations in each civilization; r is the number of good chromosomes selected for the next civilization; and c is the number of civilizations of the GA. In the first civilization, the initial population is totally generated randomly. In the other civilization, a part of the initial population is randomly generated and the rest is adopted from the previous civilization. With this structure, the proposed GA is capable of “learning” from its experience. From the second civilization, the GA has little “intelligence” so that it does not have to start the blind searching at the beginning while it still has ability to explore the wide search space. In addition, parameters of the proposed GA: population size (p),

Fig. 1 Roulette wheel representing the selection probabilities of chromosomes

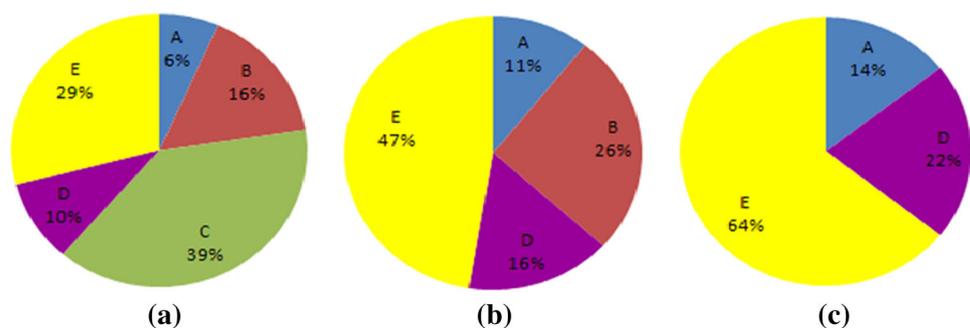
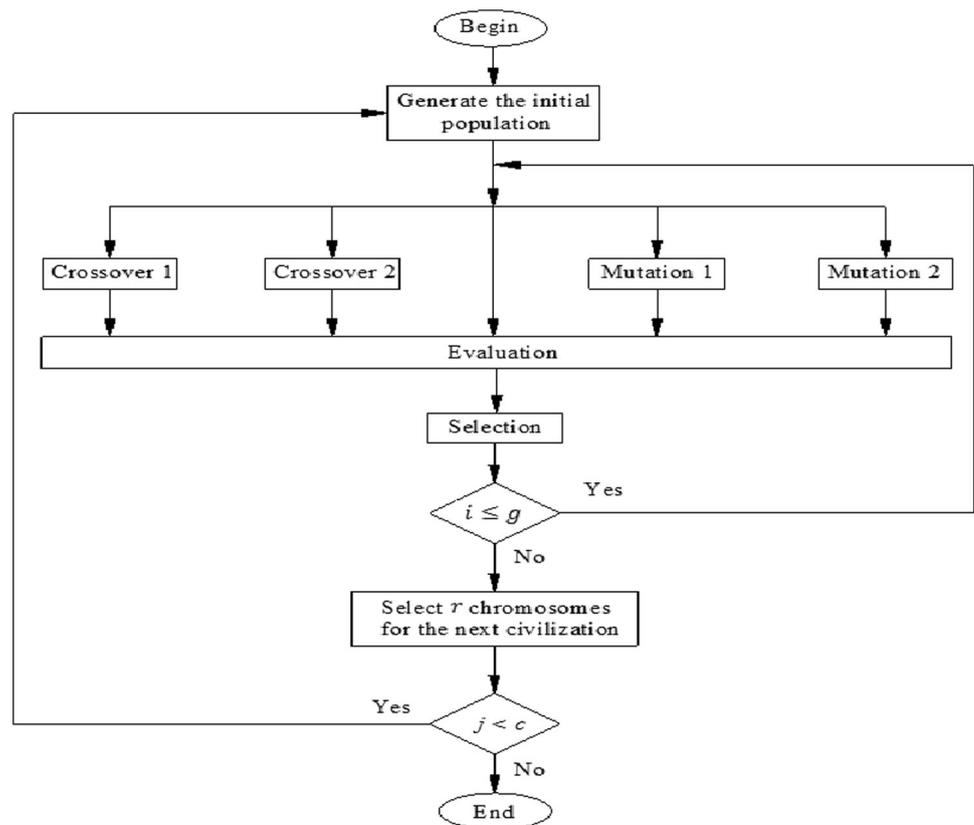


Fig. 2 Structure of the proposed GA



rate of the crossover 1 (c_1), rate of the crossover 2 (c_2), rate of the mutation 1 (m_1), rate of the mutation 2 (m_2), number of generations in each civilization (g) and number of good chromosomes selected for the next civilization (r) are tuned by Taguchi Experimental Design.

Case study

To demonstrate the capability of the proposed GA, the case study in references (Dao and Marian 2011c, 2013) is adopted here and restated as follows.

Problem description

A manufacturing company having three different production lines is able to produce 50 different products, say P1, P2,..., P50, in the next month. Information about the products is detailed in Table 12. Resources available for the next month will be 800 h of labour, 1000 kg of material, and \$1700 K of working capital.

Additionally:

- Apart from the producing cost, overheads for running a production line also costs \$300/h to be paid from the working capital.

- Any product changeover takes 0.5, 1 and 1.5 h in production line 1, 2 and 3, respectively.
- Any product made after the deadline incurs a penalty of 5% of the initial price per day of delay.
- Any product made before its deadline contributes 10 points to the company’s customer satisfaction index.
- Any product made after the deadline for more than 10 days will not be accepted by the customer and it will be returned to the company.
- Each day after the deadline of any product incurs a penalty of 1 point of customer satisfaction index.
- The proceedings from selling products will only be available for next month, so they should be ignored for the current planning horizon-current month.
- The company can select any mix of products to produce in the next month, as long as its selection contains at least 20 different ones.
- The company can work 24 h/day, 7 days/week.

The problem is to do the planning and scheduling for next month by (1) selecting what type of product to produce, (2) determining how many products in each selected type to produce, (3) allocating the selected products to which production line, and (4) selecting the manufacturing sequences of the selected products in three production lines to maximize the profit of the company as well as its

Table 12 Product information

Product	Cost (K\$)			Labour (h)			Price (K\$)	Material (kg)	Deadline (day of month)
	Line 1	Line 2	Line 3	Line 1	Line 2	Line 3			
P1	37	34	36	8	7	8	176	15	17
P2	39	45	44	5	9	12	110	16	16
P3	14	21	17	3	8	7	107	20	13
P4	19	21	18	3	6	6	125	4	10
P5	11	11	10	10	15	18	211	13	29
P6	25	32	39	8	12	17	176	1	21
P7	26	23	28	8	12	17	163	7	12
P8	30	36	36	5	6	11	132	8	25
P9	11	7	12	6	10	12	101	15	10
P10	10	6	9	9	14	15	197	11	12
P11	26	22	29	9	12	13	113	17	26
P12	28	32	26	4	7	9	175	9	5
P13	14	13	15	5	10	9	175	7	19
P14	12	19	12	2	5	10	178	15	29
P15	33	27	27	8	11	10	130	3	23
P16	25	29	35	9	12	14	133	14	26
P17	14	18	14	9	14	18	117	19	21
P18	15	9	11	9	13	15	172	4	25
P19	13	14	12	9	8	13	199	19	5
P20	16	12	7	4	7	9	159	4	14
P21	38	41	36	6	8	13	213	5	21
P22	26	21	20	2	5	4	147	3	20
P23	30	36	30	2	5	10	128	10	30
P24	12	11	12	6	8	11	102	13	5
P25	37	31	32	7	12	13	111	1	7
P26	20	22	25	5	9	12	129	19	12
P27	15	17	11	5	4	7	205	3	16
P28	40	46	46	5	4	4	148	15	19
P29	34	28	31	9	13	13	130	11	5
P30	25	25	27	8	8	13	208	12	25
P31	37	31	32	10	15	16	118	17	23
P32	13	20	15	4	3	7	215	3	9
P33	37	30	23	4	6	5	123	1	17
P34	36	43	39	10	12	14	154	12	8
P35	24	27	31	2	3	2	156	3	23
P36	14	8	5	4	4	3	148	3	25
P37	26	30	30	8	10	9	105	4	18
P38	35	40	45	2	5	9	153	7	29
P39	19	16	22	2	2	1	152	14	18
P40	27	28	32	4	8	12	158	14	15
P41	20	19	17	7	11	14	163	6	29
P42	20	19	16	4	4	8	149	11	27
P43	32	38	43	6	11	15	153	10	19
P44	22	20	14	4	3	3	215	7	22
P45	29	24	29	9	13	15	200	1	10
P46	26	26	30	8	7	10	153	13	16
P47	13	8	8	3	3	5	148	15	6

Table 12 continued

Product	Cost (K\$)			Labour (h)			Price (K\$)	Material (kg)	Deadline (day of month)
	Line 1	Line 2	Line 3	Line 1	Line 2	Line 3			
P48	26	22	15	8	8	7	164	3	29
P49	13	10	4	2	7	11	157	18	12
P50	32	31	32	6	7	9	217	20	20

Table 13 Taguchi experimental design and experimental data

Experiment	Parameters of the proposed GA							Computing time (min)	Fitness value achieved		
	<i>p</i>	<i>c1</i>	<i>c2</i>	<i>m1</i>	<i>m2</i>	<i>g</i>	<i>r</i>		Run 1	Run 2	Run 3
1	50	20	20	10	10	500	5	60	16256.8	17921.1	15892.7
2	50	20	20	10	30	1000	25	60	17321.9	18404.1	17763.7
3	50	20	20	10	50	1500	45	60	17230.8	17789.8	17012.3
4	50	50	50	30	10	500	5	60	15509.2	15655.8	15759.2
5	50	50	50	30	30	1000	25	60	17996.8	15714.3	16554.0
6	50	50	50	30	50	1500	45	60	17297.2	16231.4	18604.3
7	50	80	80	50	10	500	5	60	15999.7	16709.0	15878.1
8	50	80	80	50	30	1000	25	60	16718.3	15006.6	16305.6
9	50	80	80	50	50	1500	45	60	16091.0	16303.0	15650.0
10	100	20	50	50	10	1000	45	60	14697.6	14442.1	15288.5
11	100	20	50	50	30	1500	5	60	15131.0	14198.6	15888.5
12	100	20	50	50	50	500	25	60	15710.0	15961.7	14971.8
13	100	50	80	10	10	1000	45	60	14870.3	15733.4	16727.7
14	100	50	80	10	30	1500	5	60	17240.7	16596.5	17809.4
15	100	50	80	10	50	500	25	60	17209.0	17887.3	16596.7
16	100	80	20	30	10	1000	45	60	15324.7	13644.4	15201.7
17	100	80	20	30	30	1500	5	60	14932.1	15401.3	14749.9
18	100	80	20	30	50	500	25	60	16204.8	16688.7	15400.6
19	150	20	80	30	10	1500	25	60	15498.7	15281.2	15569.6
20	150	20	80	30	30	500	45	60	15078.4	15843.8	15813.2
21	150	20	80	30	50	1000	5	60	16434.3	16356.7	16368.5
22	150	50	20	50	10	1500	25	60	12814.5	13172.4	12406.7
23	150	50	20	50	30	500	45	60	14993.3	15577.3	13479.8
24	150	50	20	50	50	1000	5	60	14168.4	15820.5	15188.4
25	150	80	50	10	10	1500	25	60	16085.0	15823.3	15663.4
26	150	80	50	10	30	500	45	60	16935.8	15528.8	16977.9
27	150	80	50	10	50	1000	5	60	16330.4	15702.0	17072.4

customer satisfaction index while satisfying simultaneously all constraints described above. It should be noted that the weight coefficients of the two objective functions, total profit and customer satisfaction index, were assumed to be 0.7 and 0.3, respectively.

Results and discussions

The case study problem as described above was solved by the proposed GA. Taguchi Experimental Design based method was used to select the parameters of the proposed GA.

Table 14 ANOVA analysis

Source	DE	Seq SS	Adj SS	Adj MS	F	P
<i>p</i>	2	22585904	22585904	11292952	21.05	0.000
<i>c1</i>	2	945064	945064	472532	0.88	0.419
<i>c2</i>	2	5397435	5397435	2698718	5.03	0.009
<i>m1</i>	2	35590172	35590172	17795086	33.17	0.000
<i>m2</i>	2	15856076	15856076	7928038	14.78	0.000
<i>g</i>	2	730998	730998	365499	0.68	0.509
<i>r</i>	2	153189	153189	76595	0.14	0.867
Error	66	35404549	35404549	536433		
Total	80	116663388				

Table 15 Parameter set of the proposed GA

<i>p</i>	<i>c2</i>	<i>c2</i>	<i>m1</i>	<i>m2</i>	<i>g</i>	<i>r</i>
50	20	80	10	50	500	5

Taguchi Orthogonal Array Design (L27–3⁷) generated by Minitab software and the related experimental data are shown in Table 13. It is noted that to make a fair comparison, the proposed GA in any experiment was run for exactly the same computing time. In addition, to obtain the consistent experimental result, each experiment was repeated for three times. ANOVA analysis carried out in Minitab has revealed the effects of the seven parameters on the performance of the

proposed GA as shown in Table 14 and Fig. 3. Based on the results in Table 14 and Fig. 3, the parameters of the proposed GA were chosen as shown in Table 15.

Three commercial optimisation solvers: Pattern Search (PS solver), Simulated Annealing (SA solver) and Genetic Algorithm (GA solver) in Matlab were used herein as the benchmarks to verify the effectiveness of the proposed GA. The three solvers were used to solve exactly the same case study problem and their performances are compared to one of the proposed GA as shown in Table 16 and Fig. 4. It is noted that the parameters of the solvers are set by default and unit of the computing time in Table 16 is minute. It can be seen from Table 16 that the computing time of the proposed GA is 35.4, 87.7 and 43.0% smaller compared to those of PS solver, SA solver and GA solver, respectively. More importantly, the proposed GA also outperforms the

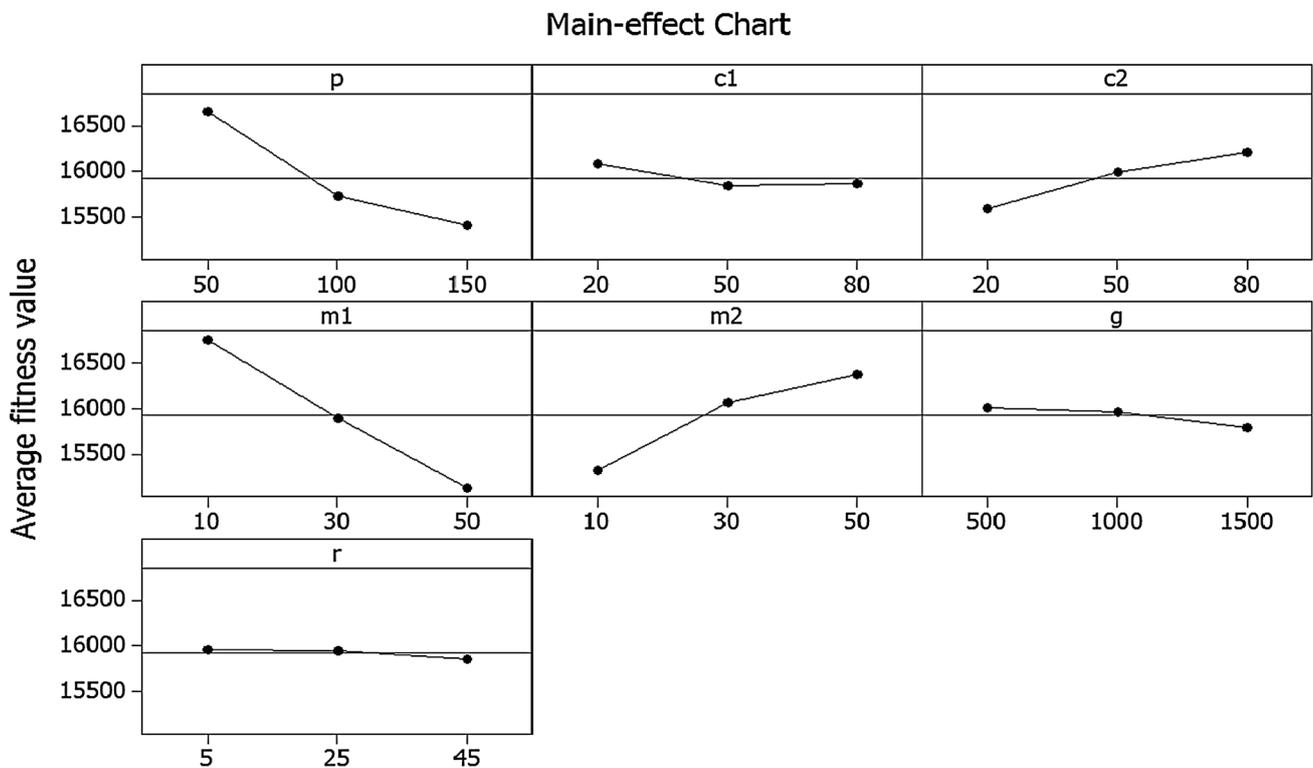
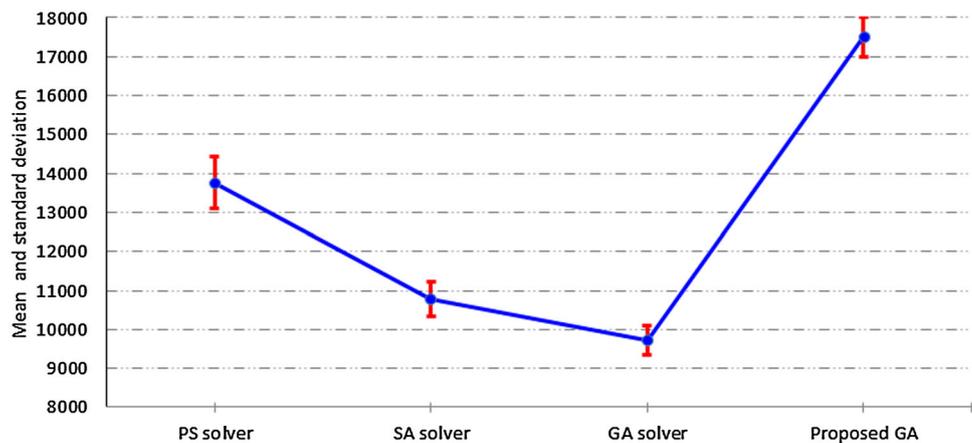


Fig. 3 Effects of the parameters on the performance of the proposed GA

Table 16 Performance comparison

Trial	No. of obj. fun. evaluations	PS solver		SA solver		GA solver		Proposed GA	
		Fitness value	Computing time						
1	798900	13654.1	107.3	11835.0	541.9	10308.7	124.4	17111.8	70.4
2	798900	12286.9	102.3	10869.0	541.3	9712.1	114.5	17675.1	64.6
3	798900	14291.6	105.1	10446.8	545.6	9417.0	114.2	17092.1	69.3
4	798900	14225.3	104.8	10765.9	545.0	10214.4	118.5	16680.9	63.0
5	798900	14050.0	105.4	10345.2	579.5	9699.4	124.6	18027.6	68.3
6	798900	12894.0	103.6	10825.0	546.6	9714.2	119.3	17949.5	75.4
7	798900	13908.7	110.7	10345.4	562.7	9886.7	121.2	18033.7	68.3
8	798900	14017.1	103.9	10556.0	553.2	9259.8	121.6	17658.9	62.6
9	798900	13894.8	102.2	11103.9	559.5	9752.3	115.7	18009.3	65.4
10	798900	14364.8	103.1	10847.6	543.2	9220.2	114.6	16908.6	70.4
Average	Std. deviation	13758.7	104.8	10794.0	551.8	9718.5	118.9	17514.7	67.8
		665.5	2.6	443.4	12.2	361.1	4.0	518.1	3.9

Fig. 4 Comparison in terms of the solution quality



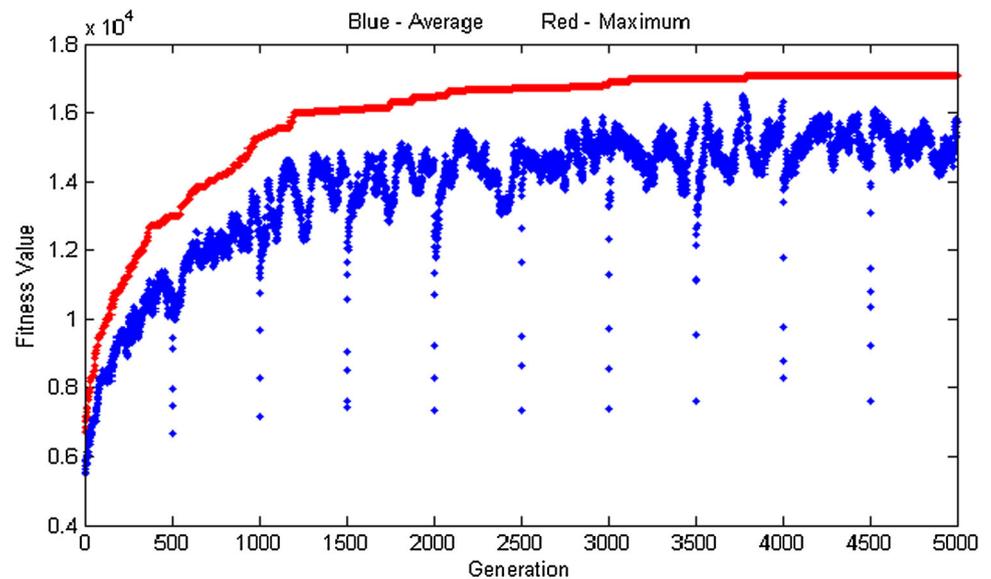
three commercial optimisation solvers in term of the solution quality. On average, the proposed GA provides 27.3, 62.3 and 80.2% better solution compared to PS solver, SA solver and GA solver, respectively. In term of consistency of the solution quality, the solvers as well as the proposed GA are about the same indicated by the standard deviation values shown in Table 16 and Fig. 4.

As can be seen from Table 16, the best solution to the case study problem found by the proposed GA is the one with the fitness value of 18033.7, which is 26.2% better than the solution already published by Dao and Marian (2011c, 2013). Detail of the best solution to the case study problem ever achieved is shown in Table 17. In addition, a typical evolution of the solution quality found by the proposed GA is shown in Fig. 5. It can be seen from Fig. 5

that the solution can be convergent to the optimal one after a few civilizations.

The experimental results revealed that the proposed structure of GA is better than the conventional one; that is because it allows the GA to “learn” for its experience. From the second civilization, the GA does not have to start the totally blind searching at the beginning because it has adopted some best chromosomes for the previous civilization. However, the question is how many chromosomes should be migrated to the next civilization. If this number is large, the exploration of the GA will be limited. As a result, the probability of getting stuck in local optimums will increase. Whereas, if the number is small, the evolution process will be slow and sometimes valuable chromosomes will be lost. In this case study problem, the

Fig. 5 A typical evolution of the solution quality found by the proposed GA



- Incorporation of stochastic events into the model and investigating their influence on the optimality.
- Developing the computing technique to shorten the processing time.
- Further validating the effectiveness of the proposed GA by comparing it with other approaches such as GAMS or other optimization software.

Acknowledgements The first author is grateful to Australian Government for sponsoring his PhD study at the University of South Australia, Australia in form of Endeavour Award.

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

References

- Abtahi AR, Bijari A (2016) A novel hybrid meta-heuristic technique applied to the well-known benchmark optimization problems. *J Ind Eng Int* 1–13 doi:[10.1007/s40092-016-0170-x](https://doi.org/10.1007/s40092-016-0170-x)
- Cadoli M et al (2000) NP-SPEC: an executable specification language for solving all problems in NP. *Comput Lang* 26(2–4):165–195
- Dao SD, Marian R (2011) Optimisation of precedence-constrained production sequencing and scheduling using genetic algorithms. In *The IAENG International Conference on Artificial Intelligence and Applications*. Hong Kong, pp 59–64
- Dao SD, Marian R (2011) Modeling and Optimization of Precedence-Constrained Production Sequencing and Scheduling Using Multi-Objective Genetic Algorithm. In *International Conference of Computational Intelligence and Intelligent Systems*. London, pp 1027–1032
- Dao SD, Marian R (2011c) Modeling and Optimisation of Precedence-Constrained Production Sequencing and Scheduling for Multiple Production Lines Using Genetic Algorithm. *Computer Technology and Application* 2(6):487–499
- Dao SD, Marian R (2013) Genetic Algorithms for integrated optimisation of precedence-constrained production sequencing and scheduling. In: Ao SI, Gelman L (eds) *Electrical engineering and intelligent systems*. Springer, New York, pp 65–80
- He K, Huang W, Jin Y (2012) An efficient deterministic heuristic for two-dimensional rectangular packing. *Comput Oper Res* 39(7):1355–1363
- Javanmard H, Koraeizadeh AAW (2016) Optimizing the preventive maintenance scheduling by genetic algorithm based on cost and reliability in National Iranian Drilling Company. *J Ind Eng Int* 12:1–8. doi:[10.1007/s40092-016-0155-9](https://doi.org/10.1007/s40092-016-0155-9)
- Kim YK, Park K, Ko J (2003) A symbiotic evolutionary algorithm for the integration of process planning and job shop scheduling. *Comput Oper Res* 30(8):1151–1171
- Liu Y, Fang Y (2012) Boost the Integration of planning and scheduling: a heuristics approach. *Proc Eng* 29:3348–3352
- Maravelias CT, Sung C (2009) Integration of production planning and scheduling: Overview, challenges and opportunities. *Comput Chem Eng* 33(12):1919–1930
- Marian R, Luong LHS, Akararungruangkul R (2008a) Optimisation of distribution networks using Genetic Algorithms. Part 1-problem modelling and automatic generation of solutions. *Int J Manuf Technol Manage* 15(1):64–83
- Marian R, Luong LHS, Akararungruangkul R (2008b) Optimisation of distribution networks using Genetic Algorithms. Part 2-the Genetic Algorithm and Genetic Operators. *Int J Manuf Technol Manage* 15(1):84–101
- Moradgholi M et al (2016) A genetic algorithm for a bi-objective mathematical model for dynamic virtual cell formation problem. *J Ind Eng Int* 12(3):343–359. doi:[10.1007/s40092-016-0151-0](https://doi.org/10.1007/s40092-016-0151-0)
- Shao X et al (2009) Integration of process planning and scheduling—a modified genetic algorithm-based approach. *Comput Oper Res* 36(6):2082–2096
- Shapiro M, Delgado-Eckert E (2012) Finding the probability of infection in an SIR network is NP-Hard. *Math Biosci* 240(2):77–84

