

# Optimisation of Precedence-constrained Production Sequencing and Scheduling Using Genetic Algorithms

Son Duy DAO & Romeo MARIAN

**Abstract**—This paper present the development of a Genetic Algorithm (GA) for optimisation of precedence-constrained production sequencing and scheduling problems. This class of problems requires a double optimisation - for sequencing and scheduling - at the same time. Due to nature of constraints, novel strategies for encoding chromosomes, crossover, mutation operations and handling constraints have been developed. The GA developed to solve this class of problems uses variable length chromosomes and its capability is demonstrated using a complex and realistic case study. The result obtained for the case study shows that the proposed GA is applicable to real-life precedence-constrained production sequencing and scheduling optimisation problems, which are ubiquitous to production and manufacturing environments.

**Index Terms**—genetic algorithm, precedence-constrained sequencing and scheduling, optimisation, cutting operation

## I. INTRODUCTION

THE precedence-constrained problems are a class of common problem facing manufacturing companies. They apply to project management, logistics, routing, assembly flow, scheduling, and networking [19]. Optimisation of precedence-constrained production sequencing and scheduling problem concerns development of the production planning by selecting what products to produce and in what order to satisfy simultaneously all of the given constraints while maximising company profit. This problem is generally described as follows: For each period of time, one company can produce a variety of products, say A1, A2, ..., An and each product has a cost to produce, a price to sell, labour requirement, material requirement and deadline. In addition, the labour, material and working capital available for production are limited. Moreover, the company has to face a number of the following constraints:

- any product changeover takes B hours;
- cost of running the company is \$C/hour and has to be paid from the working capital;
- any product made after its deadline incurs a penalty of D% of the initial price per day;
- for certain period of time, at least E different products have to be produced;
- the company can work on one product at a time.

The objective is to select what products to produce and in

Manuscript received December 2010; revised January, 2011.

Son Duy DAO is a postgraduate student at School of Advanced Manufacturing & Mechanical Engineering, University of South Australia, Australia (corresponding author, e-mail: daosd001@mymail.unisa.edu.au).

Romeo MARIAN, PhD is Senior Lecturer and Program Director at School of Advanced Manufacturing & Mechanical Engineering, University of South Australia, Australia (e-mail: romeo.marian@unisa.edu.au).

what order to maximise the total profit of the company while satisfying simultaneously all above constraints.

This class of problems can be considered as a multi-dimensional, precedence constrained, knapsack problem. The knapsack problem is a classical NP-hard problem and it has been thoroughly studied in the last few decades [21].

Although a number of production planning approaches have been developed so far, highly complex production planning or real-life planning problems have not been thoroughly solved yet and their optimal solutions are still so far from perfect due to multiplicity of constraints involved. Such constraints make the optimisation very difficult when using traditional optimisation methods and approaches. Recently, a number of researchers have applied GA to solve optimisation problems of similar complexity. However, little attention has been given to dealing with problems with solutions of variable size – accordingly, the length of chromosome encoding the solution varies.

This study uses GA with novel encoding, crossover and mutation strategies to optimise the precedence-constrained production sequencing and scheduling, which permits the search for an optimum, while considering simultaneously a large number of constraints of contradictory nature which causes the solution to have variable size. The proposed strategy is illustrated by a realistic case involving complex constraints.

## II. LITERATURE REVIEW

Sequencing and scheduling is about optimally allocating scarce resources to activities over time [6]. In 1990, Chen attempted to use an AND/OR precedence-constrained sequencing problem (PCSP) to solve assembly scheduling problem in which the precedence relations result from geometric and physical constraints of the assembled items. Similarly, Marian [12] used GA to optimise precedence-constrained assembly sequences. In order to deal with printed circuit board assembly problem, Duman & Or [5] developed an approach which initially ignores precedence relations and solves the problem as a pure travelling salesman problem (TSP), and then it is applied to eliminate component damage in the resulting TSP tour. Recently, PCSPs in disassembly scheduling problem by modification of the two-commodity network flow model was formulated by Lambert [11].

To solve scheduling model with sequence-dependent changeover cost and precedence constraints, heuristic algorithm has been developed by He & Kusiak [8]. To solve PCSP based on supply chain planning model, Moon et al. [15] proposed GA with a priority-based encoding method. A unique reasoning approach to solve PCSPs which is based

on artificial intelligent technique of case-based reasoning with evolutionary algorithm was developed by Su [18]. Recently, in the research of Yun & Moon [20], GA approach based on a topological sort-based representation procedure was proposed to solve precedence-constrained sequencing problems. It aims at locating the optimal sequence with the shortest travelling time among all feasible sequences. Clearly, this problem can be formulated as a traveling salesman problem with precedence constraints.

For single-machine sequencing with precedence constraints, Morton & Dharan [16] developed a heuristic – algorithmics - to find optimal solutions. Chekuri & Motwani [3] developed an efficient combinational 2-approximation algorithm for precedence constrained scheduling to minimize the sum of weighted completion times on a single machine. For the same problems as presented in Chekuri & Motwani [3], Azar et al. [2] considered it as a special case of the vertex cover problem and then solved it by means of vertex cover theory.

To deal with problem of scheduling a set of precedence constraint tasks onto a finite number of identical processors with and without communication overhead, Yen et al. [19] proposed priority-list scheduling method to minimize the makespan. In addition, Jonsson & Shin [10] proposed a parametrized branch-and-bound algorithm for scheduling precedence-constrained tasks on a multiprocessor system to minimize the maximum task lateness in the system. Selvakumar & Siva [17] developed heuristic algorithms to obtain near optimal schedules in a reasonable amount of computation time for scheduling precedence constrained task graphs with non-negligible intertask communication onto multiprocessors.

For scheduling precedence-constrained computations on message-passing systems, Al-Mouhamed & Al-Maasarani [1] proposed a class of global priority-based scheduling heuristics, called generalized list scheduling. Hary & Ozguner [8] used point-to-point networks to solve precedence-constrained task allocation for pipelined execution.

However, it can be also seen from the above that research attempting to optimise both sequencing and scheduling at the same time for precedence-constrained optimisation problems, especially for the large-size or real-life problems is largely missing.

In this paper, a complex and realistic mathematical model for optimisation of precedence-constrained production sequencing and scheduling is built and a GA is developed for its optimisation. A special strategy for GA implementation is developed to deal with the variability of chromosome size and feasibility of chromosome. The approach is general and applicable to real-life planning problems.

### III. RESEARCH METHODOLOGY

The tool used to solve/optimize the precedence-constrained production sequencing and scheduling problem is GA. Obviously, there are two elements to be optimized simultaneously: which products to produce and in what order. The major steps of the GA are presented as follows:

- Step 1: Chromosome encoding  
The feasible solutions for the problem are encoded as chromosomes. In general, each chromosome is a string containing the products to be produced and their producing

sequences. At first, chromosome is randomly generated as a long string of number from 1 to n, which denote the product 1 to product n, respectively. Based on the constraints in labour, material, working capital and number of products required, the length of the chromosome is cut to remove some infeasible products allocated at the end of the string. After cutting operation, the chromosome is feasible and can be used to evaluate quality of the solution further. Because of cutting operation, the length of chromosome might vary to satisfy the constraints.

- Step 2: Fitness function

Fitness function is the total profit of the company. The fitness function is calculated for each chromosome.

- Step 3: Genetic operators

A number of genetic operators will be developed such as crossover, mutation, evaluation, selection. It is noted that after each crossover and mutation operations, the feasibility of off-springs must be checked and then repaired if necessary.

- Step 4: Handling constraints

Due to number of constraints involved, a chromosome has to be tested for feasibility after generation and application of genetic operators such as crossover and mutation.

- Step 5: Optimisation implementation

The classical structure of GA is chosen to search the optimal solution in terms of total profit.

Specific and challenging for this class of problems is the handling of precedence constraints. Marian et al [13], [14] developed a framework capable to encode hard precedence relations (constraints that, violated, render the solution illegal or infeasible) as constraints. However, for this class of problems a soft approach is used, in line with the character of the problem – i.e. a precedence relation, if violated (part not ready for delivery at deadline) will incur a penalty rather than rendering the sequence and schedule infeasible. A penalty implies the respective chromosome is less likely to pass in the next generation, but still may have very valuable characteristics to pass on through the evolution process.

The proposed approach is explained in detail with the aid of the case study.

### IV. A CASE STUDY

A case study is presented here to illustrate and exemplify the concepts presented in the paper, described as follows: There is a manufacturing company producing high quality customised products from specialised castings. The company can produce a variety of products, A1, A2, ..., A40. Each product has a cost to produce, a price to sell, labour requirement, material requirement and deadlines, as listed in the Table 1.

Each month, the company has 650 h for production, 1100 kg of material and a working capital of \$400K. Additionally:

- any product changeover takes 2 h;
- the cost of running the company is \$170/h and has to be paid from the working capital;
- a product made after the deadline incurs a penalty of 5% of the initial price per day of delay;
- the company works on one product at a time;
- the proceeds 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 each month, as long as its selection contains at least 12 different ones.

It is required to do the planning for next month by selecting what products to produce and in what order to maximise the profit of the company while satisfying simultaneously all constraints above.

#### A. Modelling of the problem

##### Considering:

- A manufacturing company can produce 40 different products, with details in Table 1
- The labour, material and working capital available in next month are known.

##### Determine:

- Which products to be produced;
- What is the order to produce the selected products.

**So that the profit of the company is maximized while all constraints are simultaneously satisfied.**

##### Conditions:

- Known: The labour required for product changeover, the cost of running the company, the penalty for delays;
- The cost of running the company has to be paid from the working capital;
- 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 produce any mix of products;
- The company must produce at least 12 different products each month.

#### B. Genetic Algorithm

##### Optimisation Criteria

The optimisation criterion for this problem is the overall profit of the company, converted into the fitness function. The quality of the solution is assessed through its fitness value, computed for every single chromosome in every generation.

##### Chromosome Definition

Let numbers from 1 to 40 denote the corresponding products A1 to A40 to be produced. Each chromosome is generated by two stages.

**The first stage** is to randomly generate the general chromosome, so called initial chromosome, which looks like as shown in Table 2 in which:

- Cells C3 - AP3 represent the sequence of the products to be produced.
- Cells C4 - AP4 represent the products to be produced.

E.g., the value of 20 of the Cell H4 indicates that the product A20 will be produced the sixth, as shown in Cell H3.

**The second stage** is to cut the initial chromosome based on four constraints: labor, working capital, material, minimum number of products required. The procedure is explained in the Table 3. In this Table,

- Cells C3-AP3 represent the sequence of the products to be produced before cutting operation;
- Cells C4 - AP4 - the products to be made before chromosome cutting operation;
- Cells C5 - AP5 - the changeover time between two different products - e.g., zero in Cell D5 indicates no changeover time (product 24 made consecutively - C4 and D4). Cell K5 indicates changeover is required when producing products 8 and then 36 (Cells J4 and K4);

TABLE 1 PRODUCT INFORMATION

Product	Cost (K\$)	Price (K\$)	Labour (h)	Deadline (day of month)	Material (kg)
A1	15	575	22	30	6
A2	21	127	25	25	86
A3	22	392	31	10	55
A4	8	172	31	20	88
A5	10	543	28	25	63
A6	13	97	36	15	11
A7	11	451	14	30	6
A8	3	122	25	10	6
A9	24	413	19	25	77
A10	23	260	23	10	47
A11	3	277	27	25	73
A12	6	477	13	20	72
A13	3	557	18	5	19
A14	10	191	35	10	42
A15	3	229	33	25	22
A16	14	432	19	15	73
A17	3	529	33	5	56
A18	2	159	30	25	65
A19	2	607	24	10	59
A20	1	265	24	15	22
A21	18	228	34	25	42
A22	16	103	32	30	32
A23	5	198	24	25	56
A24	19	64	15	30	19
A25	1	294	30	15	29
A26	4	343	30	25	42
A27	8	502	14	20	34
A28	12	578	28	25	84
A29	6	616	36	30	18
A30	6	196	31	10	31
A31	22	201	15	15	83
A32	1	449	25	30	35
A33	6	565	31	5	89
A34	16	60	15	20	47
A35	15	393	23	10	65
A36	23	304	25	15	16
A37	11	354	31	20	31
A38	17	108	17	5	48
A39	16	48	22	30	68
A40	11	265	21	25	59

- Cells C6 – AP6 represent the labor required to make the corresponding products. E.g., the value of Cell G6 indicates it takes 28 hours to make A16 as shown in Cell G4 (from Table 1);
- Cells C7 – AP7 represent the material required to produce the corresponding products. E.g. the value of Cell C7 indicates the need for 6 kg of material to produce product A24 (in Cell C4);
- Cells C8 – AP8 represent the cost to produce the corresponding products. E.g. the value of Cell C8 indicates it costs \$15K to produce A24 (Cell C4);
- Cells C9 – AP9; C10 – AP10; C11 – AP11 represent the labor, material and cost consumed for corresponding products in Cells C4 – AP4. E.g. Cell C9 indicates that producing A24 consumes 15 labour hours;
- Cell AS3 denotes the minimum number of products required to be produced;
- Cell AS5 - the product changeover time in hours;
- Cell AS8 - the cost per hour of running company;
- Cell AS9 - the labor available for the next month;
- Cell AS10 - the material available for the next month;
- Cell AS11 - the working capital available;
- Cell AQ9 - the total labor required to produce products and production sequence shown in Cells C4 – AC4 and C3 – AC3, respectively. E.g. the total labor used is 645 hours - Cell AQ9. The value of this Cell must satisfy the labor constraint (less than or equal to 650);
- Cell AQ10 denotes the total amount of material required to produce the sequence shown in C4 – AA4. E.g. the amount of material used is 1095 kg (Cell AQ10). This value must satisfy the material constraint (less than or equal to 1100);
- Cell AQ11 - total cost of producing sequence in Cells C4-X4. The total working capital - \$386.65K, Cell

AQ11- satisfies the working capital constraint (less than or equal to \$400K);

- Cell AQ3 shows the number of different products from Cells C4 – X4 that satisfy all constraints. E.g. the number of different products to be made is 17 - Cell AQ3. This must satisfy the diversity constraint (minimum number of different products,  $\geq 12$ );

In this example, the chromosome is cut after Cell X4 because it satisfies all of the constraints - Table 3.

**To sum up:** the Cells C4 – X4 represent the final chromosome that is feasible. The length of this chromosome is different from the initial chromosome and different from time to time as shown in Tables 4 and 5.

#### Crossover Operation

In principle, in GA, crossover is a simple cut and swap operation. Because the length of the generated chromosome is variable, a modified crossover operation is required. In this study, to make sure the offspring chromosomes are feasible, crossover operation is not applied to feasible chromosome, but to the initial corresponding chromosome. After crossover operation, the corresponding off-springs will be cut to satisfy constraints and to ensure all offspring are feasible. In other words, the input of this crossover operation is the initial chromosomes as shown in Table 4, the initial population and the output is a set of feasible off-spring chromosomes.

The crossover operation is illustrated below:

- Assuming chromosomes 2 and 7 are selected from initial population for crossover, as shown in Table 6.
- Assuming the cut point for feasibility of chromosomes 2 and 7 are 21 and 23, respectively as shown in Table 6;
- The cut point for crossover operation must be between 1 and 21, in this case, randomly selected as 11;
- Swap the two ends as shown in Table 7;
- The two offspring obtained will be cut to satisfy constraints and guarantee feasibility - Table 8.

#### Mutation Operation

Due to a large number of constraints involved, after any mutation operation, the chromosome must be checked and repaired to guarantee feasibility. The mutation operation used in this study is as follows:

- Randomly select two chromosomes from the initial population, then randomly select two genes in two different chromosomes to swap.
- Test/repair the new chromosome to ensure feasibility.

This process is repeated to all of pairs of chromosomes involved in the mutation process. The process is illustrated in Tables 9 and 10.

Note: the lengths of two offspring after mutation might be different from those of the parent chromosomes due to feasibility constraints.

#### Evaluation Operation

The fitness function of each chromosome is total profit of the company. The total profit is calculated as follows

$$TP = I - (CP + CR + CD), \text{ where:}$$

TP – total profit of the company; I – total income; CP – total cost of producing products; CR – total cost of running company; CD – total cost associated with penalty due to products made after the deadline.

- ❖ Calculation of total income (I)

$$I = \sum_{i=1}^{40} a_i * p_i \text{ where:}$$

$p_i$  – the price to sell product  $A_i$ ;  $a_i$  – number of products  $A_i$  to be produced/sold;  $a_i = 0$  if the product  $A_i$  is not selected to produce,  $i = 1, \dots, 40$ ;

- ❖ Calculation of total cost of producing parts (CP)

$$CP = \sum_{i=1}^{40} a_i * c_i \text{ where:}$$

$c_i$  – the cost to produce product  $A_i$ ;  $a_i$  – number of products  $A_i$  to be produced;  $a_i = 0$  if the product  $A_i$  is not selected to produce,  $i = 1, \dots, 40$ .

- ❖ Calculation of total cost of running company (CR)

$$CR = 0.17 * (h_1 + h_2) \text{ where:}$$

$h_1$  – hours for producing selected products;  $h_2$  – hours for product changeover; 0.17 – cost per hour of running company.

- ❖ Calculation of total penalty cost due to products made after the deadline (CD)

$$CD = \sum_{i=1}^{40} 0.05 * t_i * a_i * p_i, \text{ where:}$$

$p_i$  – the price to sell product  $A_i$ ;  $t_i$  – the number of days which the product  $A_i$  is made after deadline;  $a_i$  – number of products  $A_i$  to be produced;  $a_i = 0$  if the product  $A_i$  is not selected to produce; 0.05 – late penalty (5% of initial price/day),  $i = 1, \dots, 40$ ;

It is noted that total labour in hours of the company is 650 per a 30-days month. Therefore, on average, each day the company runs  $k$  (hours),  $k = (650/30) = 21.27$

Based on this information and the sequence of producing products,  $t_i$  in above equation can be determined as follows

$$t_i = [(S_i + T_i) - D_i * k] / k \text{ where:}$$

$D_i$  – the deadline of product  $A_i$ ,  $k$  – average daily operation time of company,  $S_i$  – starting time for producing product  $A_i$ ;  $T_i$  – processing time for product  $A_i$ ;  $i = 1, \dots, 40$ .

#### Selection Operation

The selection procedure used in this study is *roulette wheel* approach which belongs to the fitness-proportional selection and selects a new population based on the probability distribution associated with fitness value [7].

#### Genetic Algorithm Implementation

The structure of GA used in this study is classic. The proposed GA has been successfully implemented in Matlab and run on an Intel Dual Core laptop, CPU T3400, 2.17 GHz, 3 GB RAM. For population of 100, crossover rate of 100% and mutation rate of 20%, the evolution over 2000 generations takes typically under 8 minutes. The GA is capable to find a good/best solution(s) quite quickly [12]. The quality of solution can be easily assessed by comparing different runs. Accordingly, the best solution among the ones obtained from different runs should be selected. If the number of runs is large enough, this validates the optimality of the solution. In this case study, the GA is run 100 times with different combinations of GA parameters and the results are shown in Tables 11 and 12. It can be concluded that:

- On average, the GA converges after 704 generations;
- On average, the maximum profit is \$16591.96K;
- The best solution among those solutions in Tables 11 and 12 is as highlighted in Table 12. With this solution, the total profit that company can achieve is \$17402.58K. The convergence of the GA for that solution is shown in Figure 1 and detail of the solution is shown in Table 13.

IMECS 2011

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC	AD	AE	AF	AG	AH	AI	AJ	AK	AL	AM	AN	AO	AP							
1																																																	
3	Chromosome	Sequence	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40							
4		Product	24	24	31	13	16	20	9	8	36	33	19	27	21	18	14	18	31	1	34	36	27	38	22	35	27	3	27	20	25	39	19	18	19	19	24	33	33	25	25								

TABLE 3 PROCEDURE OF CUTTING CHROMOSOME BASED ON GIVEN CONSTRAINTS (POST-PROCESSED IN EXCEL)

[illegible]

TABLE 4 INITIAL CHROMOSOMES – BEFORE CUTTING OPERATION

Sequence	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40		
Chromosome																																										
1	20	2	16	15	12	11	30	6	32	13	14	27	3	31	28	37	3	28	29	34	23	34	3	31	27	35	14	21	19	8	34	3	21	10	36	22	5	34	3	7		
2	9	31	15	2	34	15	32	35	27	22	36	5	5	2	14	7	40	17	30	28	31	37	13	9	4	24	22	26	24	19	7	31	39	7	33	35	4	20	30	30		
3	33	22	23	15	22	2	30	34	2	37	5	34	21	2	6	24	29	2	8	10	29	31	24	35	19	39	11	28	24	12	12	28	33	4	15	32	38	40	18			
4	18	14	19	31	30	40	9	16	29	7	14	33	10	14	19	15	4	21	12	23	2	25	18	26	13	12	18	22	6	6	10	5	5	8	28	22	21	38	26	29		
5	18	11	38	32	31	3	36	27	38	22	13	4	17	15	34	39	3	15	40	17	16	5	24	7	16	38	28	10	30	14	15	20	9	12	12	18	20	37	34	12		
6	38	29	35	21	3	38	27	4	39	12	38	8	1	35	34	14	5	25	17	37	35	25	36	9	11	1	5	24	13	31	26	8	12	6	23	7	34	2	34	10		
7	30	3	29	8	40	21	22	12	16	21	21	19	7	9	26	25	8	35	22	39	12	38	25	31	3	24	36	21	22	32	27	19	23	30	21	6	11	37	24	22		
8	38	11	37	23	30	28	6	39	18	4	5	39	23	20	26	10	28	11	26	38	29	39	31	21	18	26	13	18	27	33	13	27	21	7	13	5	35	27	33	38		
9	26	21	23	40	25	34	20	2	11	22	5	19	8	14	5	18	29	31	37	31	14	35	21	11	8	3	8	37	5	34	7	34	24	9	32	32	5	8	25	14		
10	11	32	25	28	28	37	6	25	20	22	10	30	11	9	35	2	2	18	4	39	26	27	14	20	2	9	34	4	13	33	25	37	35	11	38	19	36	19	24			

TABLE 5 FEASIBLE CHROMOSOMES – AFTER CUTTING OPERATION

[illegible]

TABLE 6 INITIAL POPULATION -INPUT OF CROSSOVER OPERATION

Sequence	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40		
Chromosome																																										
1	20	2	16	15	12	11	30	6	32	13	14	27	3	31	28	37	3	28	29	34	23	34	3	31	27	35	14	21	19	8	34	3	21	10	36	22	5	34	3	7		
2	9	31	15	2	34	15	32	35	27	22	36	5	5	2	14	7	40	17	30	28	31	37	13	9	4	24	22	26	24	19	7	31	39	7	33	35	4	20	30	40		
3	33	22	15	22	2	30	34	34	2	37	5	34	21	2	6	24	29	2	8	10	29	31	24	35	19	39	11	28	24	12	17	28	33	4	15	32	38	40	18			
4	18	14	19	31	30	40	9	16	29	7	14	33	10	14	19	15	4	21	12	23	2	16	18	26	13	12	18	22	6	6	10	5	5	8	28	22	21	38	26	25		
5	18	11	38	32	31	3	36	27	38	22	13	4	17	15	34	39	3	15	40	2	25	5	24	7	16	38	28	10	30	14	15	20	9	12	12	18	20	37	34	12		
6	38	29	35	21	3	38	27	4	39	12	38	8	1	35	34	14	5	25	17	37	35	25	36	9	11	1	5	24	13	31	26	8	12	6	23	7	34	2	34	10		
7	30	3	29	8	40	21	22	12	16	21	21	19	7	9	26	25	8	35	22	39	12	38	25	31	3	24	36	21	22	32	27	19	23	30	21	6	11	37	24	22		
8	38	11	27	30	23	28	6	39	18	4	5	39	23	20	26	10	28	11	26	38	28	39	31	21	18	26	13	18	27	33	13	27	21	7	13	5	35	27	33	38		
9	26	21	23	40	25	34	20	2	11	22	5	19	8	14	5	18	29	31	37	31	14	35	21	11	8	3	8	37	5	34	7	34	24	9	32	32	5	8	25	14		
10	11	32	25	28	28	37	6	25	20	12	10	30	11	9	35	2	2	18	4	39	26	17	14	20	2	9	34	4	13	33	25	27	35	11	38	19	36	19	24			

TABLE 7 CROSSOVER OPERATION

Sequence	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	
Chromosome																																									
1	20	2	16	15	12	11	30	6	32	13	14	27	3	31	28	37	3	28	29	34	23	34	3	31	27	35	14	21	19	8	34	3	21	10	36	22	5	34	3	7	
2	9	31	15	2	34	15	32	35	27	22	36	19	7	9	26	25	8	35	22	39	12	38	25	31	3	24	36	21	22	32	27	19	23	30	21	6	11	37	24	2	
3	33	22	23	15	22	2	30	34	34	2	37	5	34	21	2	6	24	29	2	8	10	29	34	24	35	19	39	21	28	24	12	12	28	33	4	15	32	38	40	1	
4	18	14	19	31	30	40	9	16	29	7	14	33	10	14	19	15	4	21	12	23	2	25	18	26	13	12	18	22	6	6	10	5	5	8	28	22	21	38	26	2	
5	18	11	38	32	31	3	36	27	48	32	13	4	17	15	34	39	3	15	40	17	16	5	24	7	16	38	28	10	30	14	15	20	9	12	12	18	20	37	34	1	
6	38	29	35	21	3	38	27	4	39	12	28	8	1	35	34	14	5	25	17	37	35	25	36	9	11	1	5	24	13	31	26	8	12	6	23	7	34	2	34	1	
7	30	3	29	8	40	21	22	12	16	21	21	5	5	2	14	7	40	17	30	28	31	37	13	9	4	24	22	26	24	19	7	31	39	7	33	35	4	20	30	3	
8	38	11	27	23	30	28	6	39	18	4	5	39	23	20	16	7	28	11	26	38	29	39	31	21	18	26	12	18	27	33	27	7	13	5	35	27	33	3	3		
9	26	21	23	40	25	34	20	2	11	22	5	19	8	14	5	18	29	31	37	31	14	35	21	11	8	3	37	5	34	7	34	24	9	32	32	5	8	25	1		
10	11	32	25	28	28	37	6	25	20	22	10	30	1	9	35	2	2	18	4	39	26	27	14	20	2	9	34	4	13	33	25	37	35	11	38	19	36	19	24		

TABLE 8 TEST AND REPAIR OFF-SPRING CHROMOSOMES (2 AND 7) FOR FEASIBILITY

[illegible]

TABLE 9 TWO PARENT CHROMOSOMES BEFORE MUTATION

Sequence	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40		
Parent																																										
1	15	23	16	16	18	30	15	25	34	6	8	38	39	11	12	34	32	15	15	12	11	40	3	27	21	5	19	29	38	22	37	40	19	15	5	34	17	31	23	20	20	
2	11	6	4	34	14	27	5	24	5	11	17	29	35	33	10	15	38	13	7	30	12	14	1	23	18	7	28	22	11	16	11	24	3	24	3	35	14	5	24	26	26	

TABLE 10 TWO OFFSPRING CHROMOSOMES AFTER MUTATION

[illegible]

## V. CONCLUSIONS AND FUTURE WORK

This paper has presented the approach to apply GA to optimise precedence-constrained production sequencing and scheduling problems. Due to very complex set of constraints - including precedence – involved, such as variability of solution size and feasibility of chromosome, new strategies for chromosome encoding, handling constraints, crossover and mutation operations have been developed. The robustness of the approach has been verified in the complex and realistic case study which contains the most important constraints currently encountered in a typical manufacturing company. The proposed method and the efficiency of the proposed algorithm can easily accommodate much larger and more complex problems in this class. The proposed GA has been extensively tested, for various combinations of the input parameters. The evolution of the output is consistently convergent towards the optimum.

Further work will be conducted in the following areas:

- Development of the algorithm for multiple production lines;
- Incorporation of stochastic events into the model and investigating their influence on the optimality;

TABLE 11 OUTPUT OF THE GA WITH DIFFERENT MUTATION RATE (POPULATION: 100, CROSSOVER: 100%)

Mutation Rate	20%		40%		60%		80%		100%	
	No. Run	Convergent Point	Maximum Profit	Convergent Point	Maximum Profit	Convergent Point	Maximum Profit	Convergent Point	Maximum Profit	Convergent Point
1	1200	16399.84	700	16076.47	500	15821.03	600	17183.97	500	16590.88
2	700	15446.48	700	16836.96	300	16904.83	1400	16918.52	800	16982.54
3	740	16551.35	1100	17122.91	600	16713.61	600	16855.53	600	17221.70
4	500	16793.99	700	16648.17	350	16406.26	800	16689.34	1800	16675.30
5	600	16760.52	700	16516.05	500	16730.77	1100	16714.56	1700	17160.45
6	900	16695.77	800	16221.67	800	16705.00	1200	16542.64	500	16846.71
7	500	16262.52	800	17129.37	1700	16392.19	500	17044.45	500	16672.85
8	500	15701.71	700	16396.01	600	15389.67	600	16635.15	500	16514.58
9	1600	16683.12	1800	16896.65	300	16769.64	1300	16319.54	300	16304.88
10	500	15720.16	700	16839.63	900	16821.76	1100	16811.60	700	16644.31
Average	774.00	16301.55	870.00	16668.39	655.00	16465.48	920.00	16771.53	790.00	16761.42

TABLE 12 OUTPUT OF THE GA WITH DIFFERENT CROSSOVER RATE (POPULATION: 100, MUTATION: 100%)

Crossover Rate	20%		40%		60%		80%		100%	
	No. Run	Convergent Point	Maximum Profit	Convergent Point	Maximum Profit	Convergent Point	Maximum Profit	Convergent Point	Maximum Profit	Convergent Point
1	400	15973.01	400	15896.76	1000	17162.78	370	16331.32	800	17199.11
2	700	16655.45	800	17280.17	800	16716.02	250	16545.91	650	16617.61
3	300	16420.48	450	16501.19	700	16810.10	500	16372.59	650	16522.72
4	700	15939.04	500	16864.20	500	16960.83	600	17402.58	1600	16990.85
5	800	16161.93	500	16942.29	500	16063.30	500	15915.97	300	16576.10
6	900	16426.89	600	16424.02	400	17275.13	1100	16829.99	300	16028.27
7	400	16234.02	500	16542.55	500	17159.08	500	16217.97	800	16000.31
8	900	15756.28	400	15878.11	350	16699.36	400	16511.31	500	16916.72
9	1100	16615.99	500	17086.23	900	16280.45	500	16793.83	500	16876.95
10	600	16674.79	800	17066.99	450	17183.45	700	16652.90	450	16558.77
Average	680.00	16285.79	545.00	16648.25	610.00	16831.05	542.00	16557.44	655.00	16628.74

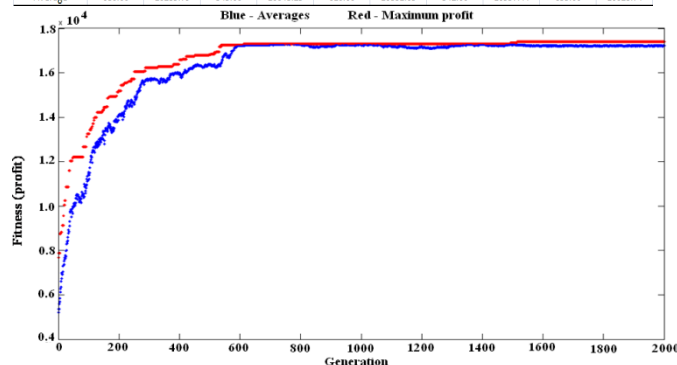


Figure 1 Evolution of the total profit – output of the GA

TABLE 13 OPTIMAL SOLUTION

Optimal solution	Sequence																																					
	Product	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13

- Improvement of the crossover and mutation operator to preserve more of the parents' genetic information;
- Development GA for extended problem, e.g. the company can produce more than one product at a time.

## ACKNOWLEDGMENT

The First Author is grateful to the Project 322, Ministry of Education and Training, Vietnam for sponsoring his study at the University of South Australia.

## REFERENCES

- [1] AL-MOUHAMED, M. and AL-MAASARANI, A. 1994. Performance evaluation of scheduling precedence-constrained computations on message-passing systems. *Parallel and Distributed Systems, IEEE Transactions on* 5, 1317-1321.
- [2] AZAR, Y., ERLEBACH, T., AMBÜHL, C. and MASTROLILLI, M. 2006. Single Machine Precedence Constrained Scheduling Is a Vertex Cover Problem. In *Algorithms – ESA 2006* Springer Berlin / Heidelberg, 28-39.
- [3] CHEKURI, C. and MOTWANI, R. 1999. Precedence constrained scheduling to minimize sum of weighted completion times on a single machine. *Discrete Applied Mathematics* 98, 29-38.
- [4] CHEN, C.L.P. 1990. AND/OR precedence constraint traveling salesman problem and its application to assembly schedule generation. In *Systems, Man and Cybernetics, 1990. Conference Proceedings., IEEE International Conference on*, 560-562.
- [5] DUMAN, E. and OR, I. 2004. Precedence constrained TSP arising in printed circuit board assembly. *International Journal of Production Research* 42, 67 - 78.
- [6] GRAVES, S.C., RINNOOY, K., A. H. G. and ZIPKIN, P.H. 1993. *Logistics of production and inventory*. North-Holland, Amsterdam.
- [7] GEN, M & Cheng, R. 1997 , Genetic algorithms and engineering design , Wiley, New York.
- [8] HARY, S.L. and OZGUNER, F. 1999. Precedence-constrained task allocation onto point-to-point networks for pipelined execution. *Parallel and Distributed Systems, IEEE Transactions on* 10, 838-851.
- [9] HE, W. and KUSIAK, A. 1992. Scheduling manufacturing systems. *Computers in Industry* 20, 163-175.
- [10] JONSSON, J. and SHIN, K.G. 1997. A parametrized branch-and-bound strategy for scheduling precedence-constrained tasks on a multiprocessor system. In *Parallel Processing, 1997., Proceedings of the 1997 International Conference on*, 158-165.
- [11] LAMBERT, A.J.D. 2006. Exact methods in optimum disassembly sequence search for problems subject to sequence dependent costs. *Omega* 34, 538-549.
- [12] MARIAN, R.M. 2003. Optimisation of assembly sequences using genetic algorithms Advanced Manufacturing and Mechanical Engineering. Adelaide, Australia: University of South Australia.
- [13] MARIAN, R.M., LUONG, L.H.S. and ABHARY, K. 2003. Assembly Sequence Planning and Optimisation Using Genetic Algorithms. Part I: Automatic Generation of Feasible Assembly Sequences. *Applied Soft Computing*, 223-253.
- [14] MARIAN, R.M., LUONG, L.H.S. and ABHARY, K. 2006. A Genetic Algorithm for the Optimisation of Assembly Sequences. *Computers and Industrial Engineering* 50, pp 503-527.
- [15] MOON, C., KIM, J., CHOI, G. and SEO, Y. 2002. An efficient genetic algorithm for the traveling salesman problem with precedence constraints. *European Journal of Operational Research* 140, 606-617.
- [16] MORTON, T.E. and DHARAN, B.G. 1978. Algorithmics for Single-Machine Sequencing with Precedence Constraints. *Management science* 24, 9.
- [17] SELVAKUMAR, S. and SIVA RAM MURTHY, C. 1994. Scheduling precedence constrained task graphs with non-negligible intertask communication onto multiprocessors. *Parallel and Distributed Systems, IEEE Transactions on* 5, 328-336.
- [18] SU, Q. 2007. Applying case-based reasoning in assembly sequence planning. *International Journal of Production Research* 45, 29-47.
- [19] YEN, C., TSENG, S.-S. and YANG, C.-T. 1995. Scheduling of precedence constrained tasks on multiprocessor systems. In *Algorithms and Architectures for Parallel Processing, 1995. ICAPP 95. IEEE First ICA/sup 3/P., IEEE First International Conference on*, 379-382 vol.371.
- [20] YUN, Y. and MOON, C. 2009. Genetic algorithm approach for precedence-constrained sequencing problems. *Journal of Intelligent Manufacturing*, 1-10.
- [21] ZOU, D., GAO, L., LI, S. and WU, J. Solving 0-1 knapsack problem by a novel global harmony search algorithm. *Applied Soft Computing In Press, Corrected Proof*.