Proceedings of the World Congress on Engineering 2011 Vol II
WCE 2011, July 6 - 8, 2011, London, U.K.

Modeling and Optimization of Precedence-Constrained Production Sequencing and Scheduling Using Multi-Objective Genetic Algorithms

Son Duy DAO & Romeo MARIAN

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

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

I. Introduction

THE precedence-constrained problem is a class of common problem facing manufacturing companies which can be found in project management, logistics, routing, assembly flow, scheduling, and networking [19]. This problem is generally described as follows. There is a manufacturing company capable to produce a variety of products for certain period of time. Each product has a cost to produce, a price to sell, a labour as well as material requirement and a deadline. In addition, the labour, material and working capital of the company available for production are limited. Any product changeover takes certain amount of time and each hour of running company also costs amount of money. Moreover, any product made after its deadline incurs a penalty of the percent of the initial price per day of delay and also affects the customer satisfaction index of the company. For any product, if number of days of delay is greater than 10, the product will be not accepted by customer and it will be returned to the company. The company also faces some other constraints such as minimum number of different products for a certain period of time, working on only one product at a time.

The problem is to select what products to produce and in what order to maximise the total profit of the company as well as customer satisfaction index while satisfying simultaneously all above constraints.

Clearly, this class of problems requires a double optimisation - for sequencing and scheduling - at the same time. It 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].

Due to a large number of complex constraints with contradictory nature involved, optimisation of highly complex production planning or real-life planning problems, especially precedence-constrained planning problems, have not been thoroughly solved yet and their optimal solutions are still so far from perfect; although a number of production planning approaches have been developed so far. Such constraints make the optimisation very difficult when using traditional optimisation methods and approaches. Recently, GA has been applied to solve optimisation problems with similar complexity. However, little attention has been given to dealing with precedence-constrained problems often resulting in solutions of variable size – accordingly, the length of chromosome encoding the solution varies from time to time.

To deal with this problem, a realistic precedence-constrained production sequencing and scheduling problem with a large number of constraints with contradictory nature which causes the solution to have variable size is firstly modelled. And then multi-objective GA with novel encoding, crossover and mutation strategies to optimise the model for two objective functions simultaneously has been developed in this paper. The proposed strategy is illustrated by a realistic case involving complex constraints.

II. Literature Review

Assembly and disassembly scheduling is one of popular precedence-constrained problems, in which the precedence relations result from geometric and physical constraints of the assembled items. Chen [4] applied AND/OR precedence-constrained sequencing approach to solve assembly scheduling problem while Marian [11] used GA to optimise precedence-constrained assembly sequences. For similar problems, 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, a precedence-constrained sequencing problem (PCSP) in disassembly scheduling was formulated by modifying the two-commodity network flow model, carried out by Lambert [10].

For PCSP in supply chain, Moon et al. [14] proposed GA with a priority-based encoding method to solve the scheduling problem. For problems with sequence-dependent changeover cost and precedence constraints, He & Kusiak [8] developed a heuristic algorithm. 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 [17]. In addition, GA approach based on a topological sort-based representation procedure was proposed to solve precedence-constrained sequencing problems, which aims

Manuscript received March 2011; revised April 2011.

Son Duy DAO was a postgraduate student at School of Advanced Manufacturing & Mechanical Engineering, University of South Australia, Australia. He is now with Thai Nguyen University of Technology, Vietnam (corresponding author to provide e-mail: daoduyson@tnut.edu.vn).

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).

at locating the optimal sequence with the shortest travelling time among all feasible sequences [20]. Clearly, this problem can be formulated as a traveling salesman problem with precedence constraints.

For single-machine sequencing with precedence constraints, Morton & Dharan [15] developed a heuristic – algoristics - to find optimal solutions while 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. In addition, Azar et al. [2] considered this problem as a special case of the vertex cover problem and then solved it by means of vertex cover theory.

For scheduling precedence-constrained tasks on a multiprocessor system, Jonsson & Shin [9] proposed a parametrized branch-and-bound algorithm to minimize the maximum task lateness in the system. In addition, Yen et al. [19] developed priority-list scheduling method to minimize the makespan in problems of scheduling a set of precedence constraint tasks onto a finite number of identical processors with and without communication overhead. Moreover, 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 was developed by Selvakumar & Siva [16].

For scheduling the precedence-constrained task allocation for pipelined execution, Hary & Ozguner [7] used point-to-point networks. Al-Mouhamed & Al-Maasarani [1] proposed a class of global priority-based scheduling heuristics, called generalized list scheduling, to schedule precedence-constrained computations on message-passing systems.

Recently, Dao & Marian [18] proposed a GA with variability of chromosome size to optimise the precedence-constrained production sequencing and scheduling for one objective function - overall profit of the company.

It is evident that optimising both sequencing and scheduling at the same time for precedence-constrained problems, especially for multi-objective function and the large-size or real-life problems, is largely missing.

In this study, a complex and realistic mathematical model for optimisation of precedence-constrained production sequencing and scheduling for not only overall profit but also customer satisfaction index is firstly developed and a multi-objective GA with special strategies to deal with the variability of chromosome size and feasibility of chromosome is proposed to optimize the model. The approach is general and applicable to real-life planning problems.

III. Research Methodology

In this paper, GA with new features in chromosome encoding, mutation and crossover operations is proposed to optimise the precedence-constrained production sequencing and scheduling problems. The major steps of the GA are presented as follows:

- Step 1: Chromosome encoding

Each chromosome, defined as a string containing the products to be produced and their producing sequences, encodes a feasible solution for the problem. In order to generate a feasible chromosome, two following steps are proposed. The first step is to generate at random a long string of number from 1 to n, which denote the product 1 to product n, respectively. It should be noted that the length of the string could be greater than n and the string generated in this step is called *initial chromosome*. The second step is to cut or remove some products allocated at the end of the *initial chromosome* based on the constraints in labour, material, working capital and number of different products required. After the cutting operation, the chromosome becomes feasible, so called *feasible chromosome*, and can be used to evaluate quality of the solution. Accordingly, the length of *feasible chromosome* might vary to satisfy all of the constraints.

- Step 2: Fitness function

Fitness function is sum of total profit of the company and customer satisfaction index. It should be noted that weight coefficient should be introduced to determine the contribution of each function: total profit or customer satisfaction index. The fitness function is calculated for each chromosome.

- Step 3: Genetic operators

Due to variable size of chromosome, new crossover and mutation operations, in which those conventional operations should not be applied to *feasible chromosome* but to *initial chromosome*, are proposed. After each crossover or mutation operation applied to *initial chromosome*, constraint-based cutting operation must be applied to make sure that the off-spring chromosomes are feasible.

- Step 4: Optimisation implementation

The proposed GA has classical structure and it is implemented in computer to search the optimal/good solutions for total profit as well as customer satisfaction index for the problem.

The most difficult task in solving this class of problems is to handle the precedence constraint. This constraint can be divided into two categories: hard precedence constraint and soft precedence constraint. Hard precedence constraint is a constraint that will make the solution illegal or infeasible if violated and a framework capable to encode hard precedence constraint has been developed by Marian et al [12], [13]. However, this study mentions the problem with soft precedence constraints, which will incur a penalty if violated rather rendering the sequence and schedule infeasible. A penalty implies that 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 illustrated by a case study.

IV. A Case Study

To illustrate the proposed approach and demonstrate its capability, a comprehensive case study is given and described as follows.

There is a manufacturing company which can produce a variety of products, P1, P2, ..., P50. Information about the products is detailed in Table 1. Moreover, there are only 250 hours of labour for production, 800 kg of material and $500K as working capital available for each month.

Additionally:

- running the company costs $200/h and this expense has to be paid from the working capital;
- any product changeover takes 1hour;
- any product made after the deadline incurs a penalty of 5% of the initial price per day of delay;

ISBN: 978-988-19251-4-5
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

- 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;
- any product made within the deadline gives 10 points of customer satisfaction index;
- each day after the deadline of a product incurs a penalty of 1 point of customer satisfaction index;
- the company can work on only one product at a time;
- 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 each month, as long as its selection contains at least 20 different ones;
- the company can work 24h/day, 7 days/week.

Problem is to do the planning for next month by selecting what products to produce and in what order to maximise the profit of the company as well as its customer satisfaction index while satisfying simultaneously all constraints above.

A. *Modelling of the problem*

Considering:

- A manufacturing company can produce 50 different products, with details in Table 1;
- Characteristics of the customer and capacity of the company such as 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:

Profit of the company as well as its customer satisfaction index is maximized while all constraints are simultaneously satisfied.

Conditions:

- The labour required for product changeover, the cost of running the company, the penalty for delays are known;
- 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 can work on only one product at a time
- The company must produce at least 20 different products in next month.

B. *Genetic Algorithm*

Optimisation Criteria

Optimisation criterions for this problem are overall profit of the company and its customer satisfaction index, converted into the fitness function. 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 50 denote the corresponding products P1 to P50. 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 region from Cell C3 to AZ4 in Table 2:

- Cells C3 - AZ3 represent the sequence of the products to be produced.
- Cells C4 - AZ4 represent the products to be produced.

E.g., the value of 45 of the Cell L4 indicates that the product P45 will be produced at the tenth, as shown in Cell L3.

TABLE 1 PRODUCT INFORMATION

Product	Cost (K$)	Price (K$)	Labour (h)	Deadline (day of month)	Material (kg)
P1	15	226	1	24	20
P2	14	294	4	7	37
P3	30	118	1	26	47
P4	23	119	3	13	3
P5	12	255	10	5	50
P6	18	132	6	11	44
P7	8	219	12	22	7
P8	22	169	2	13	20
P9	27	266	10	24	41
P10	27	257	12	26	21
P11	25	245	12	8	36
P12	4	176	8	17	13
P13	26	247	14	19	47
P14	8	254	11	23	19
P15	18	291	3	11	32
P16	12	144	12	24	46
P17	19	285	5	5	48
P18	2	110	6	17	38
P19	29	179	6	13	13
P20	15	274	3	27	41
P21	20	239	3	23	9
P22	6	127	2	20	21
P23	11	251	8	28	22
P24	3	192	15	25	11
P25	6	124	10	19	1
P26	4	135	2	13	13
P27	14	181	9	25	45
P28	14	218	15	10	28
P29	19	192	11	11	7
P30	24	243	13	16	18
P31	13	116	2	30	14
P32	16	134	5	6	27
P33	30	223	11	17	14
P34	22	270	14	6	26
P35	10	175	15	21	45
P36	26	205	12	15	26
P37	11	247	12	24	46
P38	11	152	6	14	5
P39	11	146	12	8	5
P40	18	175	1	23	7
P41	28	198	1	24	9
P42	2	244	7	30	38
P43	20	235	14	12	27
P44	19	272	11	21	21
P45	19	209	14	8	46
P46	8	226	8	9	30
P47	9	286	13	25	50
P48	3	135	1	21	32
P49	15	151	1	30	42
P50	9	247	9	14	14

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 2. In this Table,

- Cells C3-AZ3 - the sequence of the products to be produced before chromosome cutting operation;
- Cells C4 – AZ4 - the products to be made before chromosome cutting operation;
- Cells C5 – AZ5 - the changeover time between two different products - e.g., zero in Cell AP5 indicates no changeover time (product P7 made consecutively - AO4 and AP4). Cell H5 indicates changeover of one hour is required when producing products P46 and then P2 (Cells G4 and H4);
- Cells C6 – AZ6 - the labor required to make the corresponding products. E.g., the value of Cell G6 indicates that it takes 8 hours to make P46 as shown in Cell G4 (from Table 1);
- Cells C7 – AZ7 - the material required to produce the corresponding products. E.g. the value of Cell C7 indicates the need for 42 kg of material to produce product P49 as shown in Cell C4 (from Table 1);
- Cells C8 – AZ8 - the cost to produce the corresponding products. E.g. the value of Cell C8 indicates that it costs $15K to produce P49, in Cell C4 (from Table 1);
- Cell BC4 - the minimum number of different products required to be produced;
- Cell BC5 - the product changeover time in hours;
- Cell BC6 - the labor available;
- Cell BC7 - the material available;
- Cell BC8 - the working capital available;
- Cell BA6 - the total labor required to produce products and production sequence as shown in Cells C4 – AC4 and C3 – AC3, respectively. The total labor used, labors for production and changeover, is 249 hours - Cell

ISBN: 978-988-19251-4-5
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

BA6. The value of this Cell must satisfy the labor constraint - less than or equal to 250;

- Cell BA7 - the total amount of material required to produce the products with sequence shown in C4 –AC4. E.g. the amount of material used is 779 kg, shown in Cell BA7. This value must satisfy the material constraint - less than or equal to 800;
- Cell BA8 - total cost of producing sequence in Cells C4-AB4. The total working capital used, production cost and running company cost, is $483.4K. This value satisfies the working capital constraint - less than or equal to $500K;
- Cell BA4 shows the number of different products from Cells C4 – AB4 that satisfies all of the constraints. E.g. the number of different products to be made is 21 - Cell BA4. This must satisfy the diversity constraint - minimum number of different products equal to 20;

In this example, the chromosome is cut after Cell AB4 because it satisfies all of the constraints - Table 2.

To sum up: the Cells C4 – AB4 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 3-4.

Crossover Operation

Crossover, in principle, is a simple cut and swap operation. Due to the nature of constraints, 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. This study uses two-point crossover, illustrated as follows:

- Assuming chromosomes 3 and 7 are selected from *initial population* for crossover, as shown in Table 5.
- Assuming the cut points for feasibility of chromosomes 3 and 7 are 29 and 31, respectively.
- The cut points for crossover operation must be between 1 and 29, in this case, randomly selected as 16 and 25;
- Swap the two parts as shown in Table 6;
- Two offspring obtained will be cut to satisfy constraints and guarantee feasibility as shown in Table 7.

Mutation Operation

One again, due to the constraints, the modified mutation operation is not applied to feasible chromosome, but to the initial corresponding chromosome. Clearly, after any mutation operation, all of off-spring chromosomes must be checked and repaired to guarantee feasibility. The mutation operation used in this study is as follows:

- Randomly select one chromosome from the *initial population* and then randomly select two genes in the “feasible region” of the selected chromosome to swap.
- Test/repair the new chromosome to ensure feasibility.

This process is repeated to all of chromosomes involved in mutation process and illustrated in Tables 8-9.

Note: the length of the offspring after mutation might be different from the parent chromosome due to feasibility constraints.

Evaluation Operation

The fitness function of each chromosome is sum of total profit of the company and its customer satisfaction index with given weight coefficient, which is calculated as follows

$$F = w*[I - (CP + CR + CD + CRT)] + (1-w)*S$$

where: F – fitness value; 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 deadline; CRT – total cost due to returned products or not accepted by customer because they are too late; S – total points of customer satisfaction index; w – weight coefficient, assumed w = 0.7.

- Calculation of total income (*I*)

$$I = \sum_1^{50} ai * pi$$

where: *pi* – the price to sell product *Pi*; *ai* – number of products *Pi* to be produced/sold; *ai = 0* if the product *Pi* is not selected to produce, *i = 1,.. 50;*

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

$$CP = \sum_1^{50} ai * ci$$

where: *ci* – the cost to produce product *Pi*; *ai* – number of products *Pi* to be produced; *ai = 0* if the product *Pi* is not selected to produce, *i = 1,.. 50.*

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

$$CR = r*(h1+h2)$$

where: *h1* – hours for producing selected products; *h2* – hours for product changeover; *r* – cost per hour of running company.

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

$$CD = \sum_1^{n} lp * pj * tj$$

where: *pj* – the price to sell product which is in j^{th} order of producing product; *tj* – the number of days which the product in j^{th} order of producing product is made after deadline; *lp* – late penalty (5% of initial price/day); *n* – number of products to be produced; *j = 1,… n.*

Moreover, *tj* in above equation can be determined as follows

$$tj = [(Sj + Tj) - Dj * 24]/24$$

where: *Dj* – the deadline of product which is in j^{th} order of producing product, day of month; *Sj* – starting time for producing product which is in j^{th} order of producing product; *Tj* – processing time for product which is in j^{th} order of producing product; *n* – number of products to be produced; *j = 1,… n.*

- Calculation of total cost due to returned products (*CRT*)

$$CRT = \sum_1^{50} ri * pi$$

where: *ri* – number of product *Pi* to be returned; *ri = 0* if the product *Pi* is not selected to produce; *pi* – price to sell product *Pi.*

- Calculation of customer satisfaction index (*S*)

$$S = \sum_1^{n} (sc - pd * tj)$$

where: *sc* – scale of customer satisfaction index, assuming to be 10; *pd* – penalty per day of delay, assuming to be 1; *tj* – number of days which the product in j^{th} order of producing product is made after deadline and calculated as above; *n* – number of products to be produced; *j = 1,… n.* It is noted that after 10 days of delay, the corresponding customer satisfaction is assumed to be zero.

Selection Operation

In this study, *roulette wheel* approach is used to select the population for the next generations. This approach belongs to the fitness-proportional selection and selects a new population based on the probability distribution associated with fitness value [6].

TABLE 2 PROCEDURE OF CUTTING CHROMOSOME BASED ON GIVEN CONSTRAINTS (POST-PROCESSED IN EXCEL)

	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	AQ	AR	AS	AT	AU	AV	AW	AX	AY	AZ	BA	BB	BC	BD
1																																																					Used resourse			
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	41	42	43	44	45	46	47	48	49	50				
4		Product	49	30	6	18	46	2	40	47	11	45	7	4	34	2	20	31	30	14	28	46	30	49	9	8	13	43	19	9	26	2	28	45	16	25	37	39	3	50	7	7	9	46	9	5	22	9	20	32	20	9	21	>=	20	Number of different parts required
5	Changeover time		0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1		=	1	Product changeover
6	Required labor		1	13	6	6	8	4	1	13	12	14	12	3	14	4	3	2	13	11	15	8	13	1	10	2	14	14	6	10	2	4	15	14	12	10	12	12	1	9	12	12	10	8	10	10	2	10	3	5	3	10	249.0	<=	250	Labor available
7	Required material		42	18	44	38	30	37	7	50	36	46	7	3	26	37	41	14	18	19	28	30	18	42	41	20	47	27	13	41	13	37	28	46	46	1	46	5	47	14	7	7	41	30	41	50	21	41	41	27	41	41	779.0	<=	800	Material available
8	Required cost		15	24	18	2	8	14	18	9	25	19	8	23	22	14	15	13	24	8	14	8	24	15	27	22	26	20	29	27	4	14	14	19	12	6	11	11	30	9	8	8	27	8	27	12	6	27	15	16	15	27	483.4	<=	500	Working capital available

TABLE 3 INITIAL CHROMOSOMES – BEFORE CUTTING OPERATION

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	41	42	43	44	45	46	47	48	49	50
1	39	37	12	14	15	41	21	33	15	19	19	10	19	37	17	18	41	35	12	17	8	23	42	22	9	43	7	39	10	3	19	19	48	14	9	27	41	21	9	31	19	29	22	24	41	21	17	24	9	14
2	22	47	23	50	20	11	12	28	38	15	16	49	7	27	3	1	22	34	33	30	16	16	10	46	48	12	36	13	25	14	34	31	48	41	11	32	12	49	8	15	42	12	43	27	10	2	27	2	16	10
3	10	47	37	34	32	44	20	23	11	16	14	21	28	26	12	7	29	48	33	23	44	47	34	48	5	27	23	23	15	39	14	16	30	19	46	12	37	49	47	2	32	35	6	8	14	6	6	33	14	33
4	35	28	28	32	24	21	35	3	43	5	45	40	33	44	27	35	37	38	46	47	7	48	41	42	18	45	28	15	4	8	5	39	33	47	33	4	42	15	34	31	47	42	29	26	49	29	21	41	6	37
5	20	13	43	17	17	6	29	8	46	14	14	38	28	44	35	3	46	8	18	19	2	21	49	36	9	14	22	35	20	30	2	38	14	34	25	41	47	32	50	43	31	33	21	28	14	30	33	47	29	31
6	12	42	27	48	39	47	24	14	49	27	37	29	3	8	48	44	48	9	43	41	11	48	19	39	33	7	5	17	49	25	26	5	18	27	45	11	42	27	25	35	47	41	29	29	16	37	44	14	27	5
7	28	26	37	44	2	31	8	27	14	13	24	34	48	40	40	30	25	20	47	7	10	12	17	3	24	19	17	43	2	21	14	18	8	38	39	46	22	4	37	27	43	6	39	41	26	8	21	13	5	12
8	29	21	37	28	46	4	35	35	2	25	7	39	2	12	3	37	26	10	44	32	34	15	40	41	24	11	11	38	36	17	9	3	24	18	43	15	47	7	41	35	5	48	45	26	11	37	27	19	23	15
9	5	17	22	42	4	18	8	11	11	47	19	47	34	36	25	8	50	49	31	38	46	50	41	14	49	47	45	9	38	28	12	49	26	9	16	4	34	38	44	23	36	6	49	49	41	23	45	34	10	8
10	9	32	17	46	36	49	17	47	11	4	45	40	36	32	43	42	4	34	42	25	8	24	15	20	6	23	7	1	17	26	3	36	12	29	26	23	34	48	19	34	3	10	21	19	46	25	45	49	5	5

TABLE 4 FEASIBLE CHROMOSOMES – AFTER CUTTING OPERATION

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	41	42	43	44	45	46	47	48	49	50
1	39	37	12	14	15	41	21	33	15	19	19	10	19	37	17	18	41	35	12	17	8	23	42	22	9	43	7	39	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	22	47	23	50	20	11	12	28	38	15	16	49	7	27	3	1	22	34	33	30	16	16	10	46	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	10	47	37	34	32	44	20	23	11	16	14	21	28	26	12	7	29	48	33	23	44	47	34	48	5	27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	35	28	28	32	24	21	35	3	43	5	45	40	33	44	27	35	37	38	46	47	7	48	41	42	18	45	28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	20	13	43	17	17	6	29	8	46	14	14	38	28	44	35	3	46	8	18	19	2	21	49	36	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	12	42	27	48	39	47	24	14	49	27	37	29	3	8	48	44	48	9	43	41	11	48	19	39	33	7	5	17	49	25	26	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	28	26	37	44	2	31	8	27	14	13	24	34	48	40	40	30	25	20	47	7	10	12	17	3	24	19	17	43	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	29	21	37	28	46	4	35	35	2	25	7	39	2	12	3	37	26	10	44	32	34	15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	5	17	22	42	4	18	8	11	11	47	19	47	34	36	25	8	50	49	31	38	46	50	41	14	49	47	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	9	32	17	46	36	49	17	47	11	4	45	40	36	32	43	42	4	34	42	25	8	24	15	20	6	23	7	1	17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TABLE 5 INITIAL POPULATION -INPUT OF CROSSOVER

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	41	42	43	44	45	46	47	48	49	50
1	39	37	12	14	15	41	21	33	15	19	19	10	19	37	17	18	41	35	12	17	8	23	42	22	9	43	7	39	10	3	19	19	48	14	9	27	41	21	9	31	19	29	22	24	41	21	17	24	9	14
2	22	47	23	50	20	11	12	28	38	15	16	49	7	27	3	1	22	34	33	30	16	16	10	46	48	12	36	13	25	14	34	31	48	41	11	32	12	49	8	15	42	12	43	27	10	2	27	2	16	10
3	10	47	37	34	32	44	20	23	11	16	14	21	28	26	12	7	29	48	33	23	44	47	34	48	5	27	23	23	15	39	14	16	30	19	46	12	37	49	47	2	32	35	6	8	14	6	6	33	14	33
4	35	28	28	32	24	21	35	3	43	5	45	40	33	44	27	35	37	38	46	47	7	48	41	42	18	45	28	15	4	8	5	39	33	47	33	4	42	15	34	31	47	42	29	26	49	29	21	41	6	37
5	20	13	43	17	17	6	29	8	46	14	14	38	28	44	35	3	46	8	18	19	2	21	49	36	9	14	22	35	20	30	2	38	14	34	25	41	47	32	50	43	31	33	21	28	14	30	33	47	29	31
6	12	42	27	48	39	47	24	14	49	27	37	29	3	8	48	44	48	9	43	41	11	48	19	39	33	7	5	17	49	25	26	5	18	27	45	11	42	27	25	35	47	41	29	29	16	37	44	14	27	5
7	28	26	37	44	2	31	8	27	14	13	24	34	48	40	40	30	25	20	47	7	10	12	17	3	24	19	17	43	2	21	14	18	8	38	39	46	22	4	37	27	43	6	39	41	26	8	21	13	5	12
8	29	21	37	28	46	4	35	35	2	25	7	39	2	12	3	37	26	10	44	32	34	15	40	41	24	11	11	38	36	17	9	3	24	18	43	15	47	7	41	35	5	48	45	26	11	37	27	19	23	15
9	5	17	22	42	4	18	8	11	11	47	19	47	34	36	25	8	50	49	31	38	46	50	41	14	49	47	45	9	38	28	12	49	26	9	16	4	34	38	44	23	36	6	49	49	41	23	45	34	10	8
10	9	32	17	46	36	49	17	47	11	4	45	40	36	32	43	42	4	34	42	25	8	24	15	20	6	23	7	1	17	26	3	36	12	29	26	23	34	48	19	34	3	10	21	19	46	25	45	49	5	5

TABLE 6 TWO-POINT CROSSOVER OPERATION

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	41	42	43	44	45	46	47	48	49	50
1	39	37	12	14	15	41	21	33	15	19	19	10	19	37	17	18	41	35	12	17	8	23	42	22	9	43	7	39	10	3	19	19	48	14	9	27	41	21	9	31	19	29	22	24	41	21	17	24	9	14
2	22	47	23	50	20	11	12	28	38	15	16	49	7	27	3	1	22	34	33	30	16	16	10	46	48	12	36	13	25	14	34	31	48	41	11	32	12	49	8	15	42	12	43	27	10	2	27	2	16	10
3	10	47	37	34	32	44	20	23	11	16	24	34	48	40	40	30	25	20	47	7	10	47	34	48	5	27	23	23	15	39	14	16	30	19	46	12	37	49	47	2	32	35	6	8	14	6	6	33	14	33
4	35	28	28	32	24	21	35	3	43	5	45	40	33	44	27	35	37	38	46	47	7	48	41	42	18	45	28	15	4	8	5	39	33	47	33	4	42	15	34	31	47	42	29	26	49	29	21	41	6	37
5	20	13	43	17	17	6	29	8	46	14	14	38	28	44	35	3	46	8	18	19	2	21	49	36	9	14	22	35	20	30	2	38	14	34	25	41	47	32	50	43	31	33	21	28	14	30	33	47	29	31
6	12	42	27	48	39	47	24	14	49	27	37	29	3	8	48	44	48	9	43	41	11	48	19	39	33	7	5	17	49	25	26	5	18	27	45	11	42	27	25	35	47	41	29	29	16	37	44	14	27	5
7	28	26	37	44	2	31	8	27	14	13	14	21	28	26	12	7	29	48	33	23	44	12	17	3	24	19	17	43	2	21	14	18	8	38	39	46	22	4	37	27	43	6	39	41	26	8	21	13	5	12
8	29	21	37	28	46	4	35	35	2	25	7	39	2	12	3	37	26	10	44	32	34	15	40	41	24	11	11	38	36	17	9	3	24	18	43	15	47	7	41	35	5	48	45	26	11	37	27	19	23	15
9	5	17	22	42	4	18	8	11	11	47	19	47	34	36	25	8	50	49	31	38	46	50	41	14	49	47	45	9	38	28	12	49	26	9	16	4	34	38	44	23	36	6	49	49	41	23	45	34	10	8
10	9	32	17	46	36	49	17	47	11	4	45	40	36	32	43	42	4	34	42	25	8	24	15	20	6	23	7	1	17	26	3	36	12	29	26	23	34	48	19	34	3	10	21	19	46	25	45	49	5	5

TABLE 7 TEST AND REPAIR OFF-SPRING CHROMOSOMES (3 AND 7) FOR FEASIBILITY

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	41	42	43	44	45	46	47	48	49	50
1	39	37	12	14	15	41	21	33	15	19	19	10	19	37	17	18	41	35	12	17	8	23	42	22	9	43	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	22	47	23	50	20	11	12	28	38	15	16	49	7	27	3	1	22	34	33	30	16	16	10	46	48	12	36	13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	10	47	37	34	32	44	20	23	11	16	24	34	48	40	40	30	25	20	47	7	10	47	34	48	5	27	23	23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	35	28	28	32	24	21	35	3	43	5	45	40	33	44	27	35	37	38	46	47	7	48	41	42	18	45	28	15	4	8	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	20	13	43	17	17	6	29	8	46	14	14	38	28	44	35	3	46	8	18	19	2	21	49	36	9	14	22	35	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	12	42	27	48	39	47	24	14	49	27	37	29	3	8	48	44	48	9	43	41	11	48	19	39	33	7	5	17	49	25	26	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	28	26	37	44	2	31	8	27	14	13	14	21	28	26	12	7	29	48	33	23	44	12	17	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	29	21	37	28	46	4	35	35	2	25	7	39	2	12	3	37	26	10	44	32	34	15	40	41	24	11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	5	17	22	42	4	18	8	11	11	47	19	47	34	36	25	8	50	49	31	38	46	50	41	14	49	47	45	9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	9	32	17	46	36	49	17	47	11	4	45	40	36	32	43	42	4	34	42	25	8	24	15	20	6	23	7	1	17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TABLE 8 MUTATION OPERATION APPLIED TO INITIAL CHROMOSOME

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	41	42	43	44	45	46	47	48	49	50
Parent	39	37	12	14	15	41	21	33	15	15	19	10	19	37	17	18	41	35	12	17	8	23	42	22	9	43	7	39	10	3	19	19	48	14	9	27	41	21	9	31	19	29	22	24	41	21	17	24	9	14
Offspring	39	37	12	14	15	41	21	33	12	15	19	10	19	37	17	18	41	35	15	17	8	23	42	22	9	43	7	39	10	3	19	19	48	14	9	27	41	21	9	31	19	29	22	24	41	21	17	24	9	14

TABLE 9 CUTTING OPERATION APPLIED AFTER MUTATION OPERATION

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	41	42	43	44	45	46	47	48	49	50
Parent	39	37	12	14	15	41	21	33	15	15	19	10	19	37	17	18	41	35	12	17	8	23	42	22	9	43	7	39	10	3	19	19	48	14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Offspring	39	37	12	14	15	41	21	33	12	15	19	10	19	37	17	18	41	35	15	17	8	23	42	22	9	43	7	39	10	3	19	19	48	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Genetic Algorithm Implementation

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 10% and mutation rate of 10%, the evolution over 2000 generations takes typically under 8 minutes. It is evident that GA cannot guarantee to find best solution after only one run. However, it is very good at finding good/best solution(s) reasonably quickly [11]. Therefore, it is easy to evaluate the quality of solution by comparing different runs. Accordingly, the best solution among the ones obtained from different runs should be selected. If number of runs is large enough, this validates the optimality of the solution. In this case study, the GA is run for 100 times with different combinations of GA parameters and results are shown in Tables 10 -11. During the run of the GA in this case study, it was found that for better consistent convergence, mutation as well as crossover rate should not be too high. Based on the results in tables 10-11, it can be concluded that:

- On average, the GA converges after 1322 generations;
- On average, the maximum fitness value is 6065.12;
- The best solution among those solutions in Tables 10-11 is as highlighted in Table 10. With this solution, the achieved fitness value is 6560.28. The convergence of the GA for that solution is shown in Fig. 1 and detail of the solution is shown in Table 12.

TABLE 10 OUTPUT OF THE GA WITH DIFFERENT MUTATION RATE (POPULATION: 100, CROSSOVER: 10%)

Mutation Rate	10%		20%		30%		40%		50%	
No. Run	Convergent Point	Maximum Fitness Value	Convergent Point	Maximum Fitness Value	Convergent Point	Maximum Fitness Value	Convergent Point	Maximum Fitness Value	Convergent Point	Maximum Fitness Value
1	1700	6142.26	1000	5911.40	1250	5986.02	500	5882.48	800	5831.44
2	900	5862.76	1400	6264.52	1500	6164.54	1600	6231.36	1500	6069.32
3	950	6095.16	800	6154.88	1350	6086.00	1000	6208.00	1800	6148.34
4	900	5959.08	1600	6312.94	900	6205.12	950	5927.76	1000	6338.38
5	500	5644.42	1700	5971.30	1100	6140.02	1800	6560.28	1500	6018.68
6	1500	6161.42	1100	5874.28	1200	6042.88	1000	6189.70	1200	6356.20
7	800	5896.80	1150	6248.48	850	6213.86	1200	6095.42	1850	6013.00
8	750	5661.66	1600	5961.14	1650	6485.16	1550	6030.86	1900	6004.98
9	1000	5926.92	1300	6093.84	1300	6003.42	1000	5993.14	1250	5831.44
10	1500	5921.90	1100	5967.12	1750	6154.60	900	5959.26	1150	6251.54
Average	1050	5927.24	1275	6075.99	1285	6148.16	1150	6107.83	1395	6086.33

TABLE 11 OUTPUT OF THE GA WITH DIFFERENT CROSSOVER RATE (POPULATION: 100, MUTATION: 10%)

Crossover Rate	10%		20%		30%		40%		50%	
No. Run	Convergent Point	Maximum Fitness Value	Convergent Point	Maximum Fitness Value	Convergent Point	Maximum Fitness Value	Convergent Point	Maximum Fitness Value	Convergent Point	Maximum Fitness Value
1	1100	5940.18	1500	5938.54	1700	6387.50	1300	5843.00	1800	6405.94
2	1300	5996.34	1300	5829.12	1250	6152.18	550	5916.42	1200	6159.74
3	1700	6062.12	1700	5845.50	800	6085.86	1800	6037.38	1200	6071.56
4	1500	5935.78	1650	6296.88	1800	6121.02	1900	6506.44	1800	6254.60
5	600	6097.14	1550	5962.26	1500	6012.76	450	5972.02	1450	6093.94
6	1500	5956.96	1850	6277.02	1600	6064.20	1750	5778.68	600	6044.22
7	1300	6077.30	1350	5855.28	1700	5956.10	1820	6100.84	1350	6068.78
8	1150	6027.68	800	5969.34	1200	6204.74	1850	6053.34	1850	6376.22
9	1350	6019.14	1450	5996.14	1750	6033.64	500	5873.04	1400	6034.68
10	800	5722.92	1900	5980.20	1850	6188.68	1750	6253.24	1850	6219.58
Average	1230	5983.56	1505	5995.03	1515	6120.67	1367	6033.44	1450	6172.93

TABLE 12 OPTIMAL SOLUTION

Optimal solution	Sequence	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
	Product	25	7	24	22	26	26	12	12	12	2	42	38	26	26	26	21	21	15	1	40

Optimal solution	Sequence	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40
	Product	14	21	21	21	21	21	50	50	50	41	47	48	26	26	26	26	26	26	26	26

Optimal solution	Sequence	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60
	Product	26	26	26	26	26	26	26	26	26	26	26	23	12	31	0	0	0	0	0	0

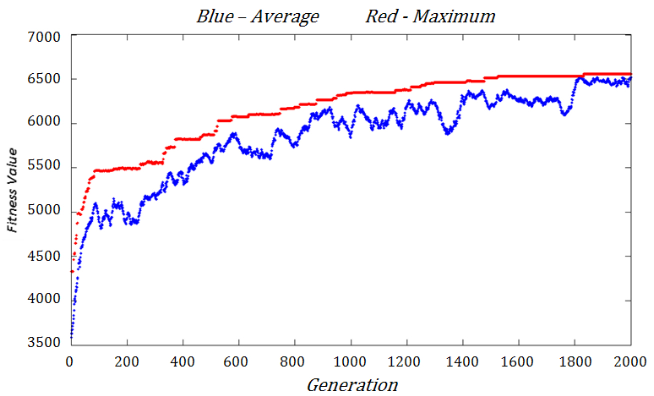

Fig. 1. Evolution of fitness value – output of the GA

V. CONCLUSIONS AND FUTURE WORK

In this paper, a class of soft precedence-constrained production sequencing and scheduling problems has been modelled. To optimise the model for two objective functions at the same time, multi objective GA has been chosen for its capability to deal with complex or real life problems. Due to the nature of constraints, the chromosome encoded the solution for the problem has variable size. Therefore, new strategies for chromosome encoding, feasibility of chromosome, crossover as well as mutation operation have been developed.

The proposed approach has been illustrated and its robustness 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 GA has been extensively tested, for various combinations of the input parameters. The evolution of the output is consistently convergent towards the optimum. It is evident that the proposed method can easily accommodate much larger and more complex problems in this class of problem.

Further work should be conducted in the following areas:

- Developing the GA for extended problem, e.g. for multiple production lines or company can produce more than one product at a time;
- Incorporation of stochastic events into the model and investigating their influence on the optimality.

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, 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] Gen, M & Cheng, R, 1997 , Genetic algorithms and engineering design , *Wiley, New York.*

[7] 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.

[8] HE, W. and KUSIAK, A. 1992. Scheduling manufacturing systems. *Computers in Industry 20*, 163-175.

[9] 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.

[10] LAMBERT, A.J.D. 2006. Exact methods in optimum disassembly sequence search for problems subject to sequence dependent costs. *Omega 34*, 538-549.

[11] MARIAN, R.M. 2003. Optimisation of assembly sequences using genetic algorithms Advanced Manufacturing and Mechanical Engineering. Adelaide, Australia: University of South Australia.

[12] 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.

[13] 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.

[14] 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.

[15] MORTON, T.E. and DHARAN, B.G. 1978. Algoristics for Single-Machine Sequencing with Precedence Constraints. *Management science 24*, 9.

[16] 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.

[17] SU, Q. 2007. Applying case-based reasoning in assembly sequence planning. *International Journal of Production Research 45*, 29-47.

[18] DAO, SD & MARIAN, R 2011, 'Optimisation of Precedence-Constrained Production Sequencing and Scheduling Using Genetic Algorithms', *Proceeding of the IAENG International Conference on Artificial Intelligence and Applications*, Hong Kong.

[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/PP., 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*.

ISBN: 978-988-19251-4-5
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)