



<b>Title</b>	<b>An introduction of dominant genes in genetic algorithm for scheduling of FMS</b>
<b>Author(s)</b>	<b>Chan, FTS; Chung, SH; Chan, PLY</b>
<b>Citation</b>	<b>IEEE International Symposium on Intelligent Control Proceedings, the 13th Mediterrean Conference on Control and Automation Proceedings, Limassol, Cyprus, 27-29 June, 2005, p. 1429-1434</b>
<b>Issued Date</b>	<b>2005</b>
<b>URL</b>	<b><a href="http://hdl.handle.net/10722/46552">http://hdl.handle.net/10722/46552</a></b>
<b>Rights</b>	<b>Creative Commons: Attribution 3.0 Hong Kong License</b>

# An Introduction of Dominant Genes in Genetic Algorithm for Scheduling of FMS

Felix T.S. Chan, S.H. Chung, and P.L.Y. Chan  
Department of Industrial and Manufacturing Systems Engineering  
The University of Hong Kong, Pokfulam Road, Hong Kong  
E-mail: ftschan@hkucc.hku.hk

**Abstract**—This paper proposed a new idea named Dominant Genes (DGs) in Genetic Algorithms (GAs) to deal with FMS scheduling problem with alternative production routing. In traditional GAs approach, the crossover mechanism will randomly select a number of genes to undergo crossover. However, these selected genes may not contain or contain only part of the critical structure of its original chromosome. In addition, since the inherited complexity of the scheduling nature, the changes in the structure of the selected genes will further influence its strength. To tackle this problem, the proposed DGs in this paper are to identify and record the best genes in the chromosome. A new crossover mechanism is also designed to ensure the best genes will undergo crossover, and retain the originality of the structure of the crossover genes. The performance of the proposed DGs is testified by comparing it with other heuristic optimizations. The comparison shows that DGs perform better than other approaches.

## I. INTRODUCTION

To factories, job shop scheduling is one of the most important tasks. A good schedule can increase the efficiency of the manufacturing systems. However, because of the inherited complexity of the problem nature and the governed constraints, determination of a good schedule is a difficult and time consuming activity. It usually involves heavy computation efforts. In many practical cases, scheduling problems is classified as NP-Hard. The computation effort grows even exponentially with the

Manuscript received December 1, 2004.

F. T. S. Chan is with the Department of Industrial & Manufacturing Systems Engineering, The University of Hong Kong, Pokfulam, Hong Kong (phone: +852 28597059; fax: +852 28586535; e-mail: ftschan@hkucc.hku.hk).

S. H. Chung is with Department of Industrial & Manufacturing Systems Engineering, The University of Hong Kong, Pokfulam, Hong Kong (e-mail: h9840627@hkusua.hku.hk).

P. L. Y. Chan is with the Department of Industrial & Manufacturing Systems Engineering, The University of Hong Kong, Pokfulam, Hong Kong (phone: +852 28592588; fax: +852 28586535; e-mail: plychan@hkucc.hku.hk).

increment of the problem size. Using pure mathematical optimization approach to determine an optimal solution may not be efficient in practice. Instead, obtaining a near optimal solution in a relatively shorter period by heuristic methodology is more appreciated.

Many different heuristic methodologies have been proposed by researchers for the past few decades, for example dispatching rules such as first in first out, shortest processing time, critical ratio, etc. In recent years, many other heuristic methodologies have been widely adopted, such as branch and bound, hill-climbing, simulated annealing, Tabu search, and especially Genetic Algorithms (GAs). This paper will mainly discuss on the application of GAs approach in dealing with scheduling problem.

Many literatures can be found in the study of GAs. Cheung et al. [1] have given a detailed tutorial survey on papers using GAs to solve classical Job-Shop scheduling problems (JSP) in their Part I survey. In Part II, they reviewed papers using hybrid GA to tackle JSP [2]. Jain and ElMarachy [3] proposed a GA to solve single process plan scheduling (SPPS) problems. Cavalieri and Gaiardelli [4] applied a hybrid GA, which combines GA with dispatching rule (Earliest Due Date), to solve multi-objective scheduling problems. Sakawa [5] combined GA with fuzzy logic to model the uncertainties of production lead time and order due date in scheduling problem. More references could be found, for example, Mori and Tseng [6], Jawahar et al., [7], Ghedjati [8].

GAs is recognized as an appropriate and efficient approach to solve scheduling problems by many researchers. However, to maximize the performance of the genetic search, an optimal set of genetic parameters will have to be determined, such as population size, crossover rate, mutation rate, generation gap, scaling window, and selection strategy [6, 9-10]. Different genetic parameters setting in different problems will lead to different performance since the setting of each parameter has its function, for example high crossover rate can increase the global searching, while low crossover rate favors fine local searching. Determining an optimal set of parameters is critical to the performance,

but is difficult to find due to the large number of possible combination.

To strategically strengthen the genetic search in different phases of evolution, many researchers proposed different kinds of Adaptive Genetic Algorithms for their particular problems. During the genetic evolution, the genetic parameters, such as population size, crossover rate, and mutation rate will change strategically [9, 11-12]. Michalewicz [9] proposed a non-uniform mutation which allows the operator to search through the solution space uniformly in the beginning stages to prevent pre-maturity of the solution pool, and then locally in the later stages for fine local tuning. Cavaliefi and Gaiardelli [11] adopted a dynamic population size approach to replace the old chromosomes with new ones to maintain the diversity of solution pool.

In many cases, researchers will usually be required to design a new chromosome to represent the possible solution to the special needs in their problems. Accordingly, many different kinds of crossover and mutation mechanism have also been specially designed to cooperate with the chromosome in order to enhance their genetic search. In scheduling problem, because of the inherited complexity, it is crucial to prevent the dramatic changes of chromosome after crossover.

The objective of this paper is to propose a new idea named Dominant Genes (DGs) in GA to solve scheduling problem of FMS with alternative production routing. The function of DGs is to represent the strong genes in the chromosome. With the proposed DGs, a new crossover mechanism is designed to enhance the performance of the genetic search. This paper will be divided into the following sections. In section II, the scheduling problem will be modeled. Section III will present the proposed DGs, and its crossover mechanism. Section VI will analyze and discuss the performance of the DGs in FMS environment, and the paper will be concluded in Section V.

## II. PROBLEM DESCRIPTION

The FMS scheduling problem is expressed in the following notation.

$\delta_{ijkh}$  Binary Integer, defined as 1 if the operation  $j$  of job  $i$  occupied time slot  $k$  on machine  $h$ , otherwise 0.

$\omega_i$  Weighting of job  $i$ .

$N$  Number of jobs.

$N_i$  Number of operations of job  $i$ .

$H$  Number of machines.

$K$  Time horizon under consideration.

$t_{ijh}$  Processing time of operation  $j$  of job  $i$  on machine  $h$ .

$S_{ij}$  Starting time of operation  $j$  of job  $i$ .

$E_{ij}$  Ending time of operation  $j$  of job  $i$ .

$D_i$  Due Date of job  $i$

$C_i$  Completion time of job  $i$ .

$T_i$  Tardiness of job  $i$ , defined as  $\max(0, C_i - D_i)$ .

In the problem, it is assumed that  $\omega_i, D_i, N, N_i, H$ , and  $t_{ijh}$  are given. Each job's operation can be processed on more than one suitable machine, but not all. The decision variable is the occupation time slot  $k$  of operation  $j$  of job  $i$  on machine  $h$  ( $\delta_{ijkh}$ ). With the solution of  $\delta_{ijkh}$  obtained, the value of  $S_{ij}, E_{ij}, D_i, C_i$ , and  $T_i$  can be calculated. The objective is to minimize:

(i) Makespan (for Example 1 in section IV-A).

$$\text{Objective Z: } \min(\max[C_i]) \quad (1.1)$$

(ii) Weighted quadratic tardiness function of the jobs (for Example 2 in section IV-B).

$$\text{Objective Z: } \min\left(\sum_i \omega_i T_i^2\right) \quad (1.2)$$

In this function, each job will have a weighting to model its importance. In addition, as the tardiness of a job increases, this job will become more critical. The problem will be subject to the following constraints:

*Precedence constraints:*

$$S_{ij} \leq E_{i(j-1)} \quad (i = 1, 2, \dots, N; j = 1, 2, \dots, N_i) \quad (2)$$

*Processing time constraints:*

$$E_{ij} - S_{ij} + 1 = t_{ijh} \quad (i = 1, 2, \dots, N; j = 1, 2, \dots, N_i; h = 1, 2, \dots, H) \quad (3)$$

*Operation constraints:*

$$\sum_{kh} \delta_{ijkh} \geq 1 \quad (i = 1, 2, \dots, N; j = 1, 2, \dots, N_i) \quad (4)$$

*Processing operation constraints:*

$$\sum_h \delta_{ijkh} \leq 1 \quad (i = 1, 2, \dots, N; j = 1, 2, \dots, N_i; k = 1, 2, \dots, K) \quad (5)$$

*Machine capacity constraints:*

$$\sum_{i,j} \delta_{ijkh} \leq 1 \quad (k = 1, 2, \dots, K; h = 1, 2, \dots, H) \quad (6)$$

In the above constraints, constraints (2) defines that the starting time of each operation can only be processed after its precedence operation. Constraints (3) defines that once an operation starts, it will be finished without interruption. Constraints (4) forces each operation to be carried out on at

least one machine throughout the horizon. Constraints (5) forces each operation to be only carried out on one machine at each time unit, and constraints (6) forces each machine to carry out only one operation at each time unit.

### III. PROPOSED DOMINANT GENES IN GENETIC ALGORITHM

#### A. Dominant Genes

Solving scheduling problem with GAs approach has been recognized by many researchers as an appropriate and efficiency approach. This paper proposed a new idea named Dominant Genes (DGs) to enhance the performance of the genetic search, specially designed to deal with the inherited problem nature of scheduling problems. The function of DGs is to identify and record the best genes in the chromosome. In the initial pool, some genes are randomly assigned as DGs, which may contain more than one. During evolutions, only those DGs undergo crossover in each pair of parents to generate a pair of offspring. Each offspring reserves most of the genes from one of the parents and inherits only the DGs from another parent. If these inherited DGs make the offspring stronger (increase fitness value) than the parent, they will become DGs in the offspring, otherwise they will become normal genes. This idea ensures that the best genes will be passed to the offspring. A normal gene can also become a DG through two types of mutations discussed in the section III-D.

#### B. Encoding of chromosome

Each gene consists of four parameters, representing Machine, Job, Operation, and Domination (MJOD). Fig. 1a shows a sample encoding of a chromosome for the scheduling of 3 jobs (each with 3 operations) on 3 machines.

1111-**2330**-2120-2130-2210-1220-3230-1311-3320  
Fig. 1a. A sample encoding of chromosome

In Fig. 1a., the second gene (2330) represents that O3 of J3 is allocated on M2, and it is not a DG denoted by 0 (DG denoted by 1). The scheduling priority of jobs on machines is defined by the ordering, from the highest priority on the

left to the lowest on the right. Therefore, Fig. 1a. indicates that O3 of J3 (the second gene: 2330) will be scheduled before O2 of J1 (the third gene: 2120) on M2.

This encoding also allows to model in alternative routing problems. Assuming O3 of J3 can also be performed on M3, then the second gene can be represented as (3330) as shown in Fig. 1b.

1111-**3330**-2120-2130-2210-1220-3230-1311-3320  
Fig. 1b. A sample encoding of alternative routing

#### C. Dominant gene crossover

In traditional GA approach, a number of genes will be randomly selected governed by a predefined crossover rate(s). However, these selected crossover genes may or may not be critical even in its original chromosome structure. In addition, it is difficult to ensure that the whole set of best genes can be selected and inherited to its offspring. Another problem is the originality of the parent's chromosome structure. After crossover, the structure of the offspring should not have dramatic changes from its parent because this will make the genetic search become a random search approach. In the DGs approach, crossover rate depends on the number of DGs in each chromosome. DGs crossover can be divided into 2 cases. If there are no DGs conflicting at the same location of the two parents, and if there are no identical jobs dominant in both parents, it will be classified as Case A as the parent chromosomes shown in Fig. 2, otherwise Case B as shown in Fig. 3a, and 3b.

#### Case A

Crossover in Case A is carried out in four steps. Fig. 2 shows a sample crossover.

- Step 1: Offspring 1 (Of1) reserves DGs from P1, and these genes will be emptied in P1.
- Step 2: Of1 inherits DGs from P2, and the genes with the same number of J and O in P1 will be emptied.
- Step 3: Copy the remaining genes with the same number of J equal to the DGs from P2 to replace those in P1.
- Step 4: Copy the non-empty genes from P1 to the non-empty genes in Of1.

#### Case A

	<b>P1:</b>	1111	-	2330	-	2120	-	2130	-	2210	-	1220	-	3230	-	<b>1311</b>	-	3320
	<b>P2:</b>	2130	-	2120	-	1110	-	2210	-	<b>1221</b>	-	3230	-	1310	-	3320	-	2330
<b>Step</b>																		
<b>1</b>	<b>Of1:</b>	1111	-	0000	-	0000	-	0000	-	0000	-	0000	-	0000	-	<b>1311</b>	-	0000
<b>2</b>	<b>Of1:</b>	1111	-	0000	-	0000	-	0000	-	<b>1221</b>	-	0000	-	0000	-	<b>1311</b>	-	0000
<b>3</b>	<b>P1:</b>	0000	-	2330	-	2120	-	2130	-	0000	-	2210	-	3230	-	0000	-	3320
<b>4</b>	<b>Of1:</b>	1111	-	2330	-	2120	-	2130	-	<b>1221</b>	-	2210	-	3230	-	<b>1311</b>	-	3320
<b>Step</b>																		
<b>1</b>	<b>Of2:</b>	0000	-	0000	-	0000	-	0000	-	<b>1221</b>	-	0000	-	0000	-	0000	-	0000
<b>2</b>	<b>Of2:</b>	1111	-	0000	-	0000	-	0000	-	<b>1221</b>	-	0000	-	0000	-	<b>1311</b>	-	0000
<b>3</b>	<b>P2:</b>	0000	-	2120	-	2130	-	2210	-	0000	-	3230	-	2330	-	0000	-	3320
<b>4</b>	<b>Of2:</b>	1110	-	2120	-	2130	-	2210	-	<b>1221</b>	-	3230	-	2330	-	1310	-	3320

Fig. 2 Dominant Gene crossover of Case A

Similar steps will be carried out for Of2 except in Step 1 where Of2 reserves DGs from P2, then inherits from P1 in Step 2. In Step 3, the remaining genes will be replaced from P1 to P2, and the non-empty genes will be copied from P2 to Of2.

One of the advantages of this crossover mechanism is that the best genes will undergo crossover. In addition, these genes will be testified whether they can increase the fitness value. If they make contribution to the chromosome, they will be identified, recorded, and inherited to its offspring. Assuming that the fitness value of Of1 is better than P1, then the inherited DGs from P2 will remain its domination. However, if it is weaker than its parents, like Of2 is weaker than P2, the inherited DGs from P1 will become normal genes.

#### Case B

In Case B, since P1 and P2 have DGs conflicting in the same location(s) or job(s), as shown in Fig. 3a and 3b respectively, a selection is required to determine which DG is stronger. However, for example, if each parent has 2 DGs conflicting, there will be 4 possible combinations. To prevent too many possible combinations, only two choices will be formed. The first choice is formed by changing all the DGs in P1 into normal genes, while the second choice by changing all those in P2 into normal genes. For example, the parents in Fig 3a will convert into the form that satisfies Case A, as shown in Fig 3c and 3d to produce Of1A, and Of1B respectively. In Fig 3c, the conflicted DG (2131) of P2 is changed to normal gene (2130) in Fig 3a. The crossover approach is that Case A can be carried out to obtain Of1A. Similarly, Of1B can be obtained from the form in Fig 3d. The stronger one (Of1A or

Of1B) will be selected as Of1. Similar procedures will be carried out to obtain Of2A, and Of2B, except that the reserved genes will be in P2, and inherited genes will be from P1.

#### D. Mutation operator

There are two types of mutation. In Mutation 1, a pair of genes will be randomly selected and swapped, as shown in Fig.4a. The purpose of this mutation is to reschedule the scheduling priority of job's operations. For example, Fig. 4a shows that after the swap, the production priority on M2 is that O3 of J3 (2330) is rescheduled to be produced after O2 and O3 of J1 (2120 and 2130).

In Mutation 2, some genes will be randomly selected and mutated, as shown in Fig. 4b. The number of mutated genes is governed by the predefined mutation rate(s). The selected gene will randomly change in the M parameter. The purpose of this mutation is to increase the diversity of the chromosomes. In both mutation operators, if Of1 is stronger than P1, the mutated gene will become DG.

#### E. Elitist strategy

To prevent the loss of the best chromosome during evolutions, the best chromosome will be identified and recorded. If the best chromosome is lost or becomes weaker after evolution, it will be inserted back into mating pool for the next evolution.

### IV. ANALYSIS AND DISCUSSION ON THE PERFORMANCE OF DG

The objective of this section is to testify the performance of the proposed DGs approach in solving FMS scheduling

#### Case B

**P1:** 1111 - 2330 - 2120 - 2130 - 2210 - 1220 - 3230 - **1311** - 3320

**P2:** **2131** - 2120 - 1110 - 2210 - 1220 - 3230 - 1310 - 3320 - 2330

Fig. 3a Dominant genes conflicting in the same location

**P1:** 1111 - 2330 - 2120 - 2130 - 2210 - 1220 - 3230 - **1311** - 3320

**P2:** 2130 - **2121** - 1110 - 2210 - 1220 - 3230 - 1310 - 3320 - 2330

Fig. 3b Dominant genes conflicting in the same job

**P1:** 1111 - 2330 - 2120 - 2130 - 2210 - 1220 - 3230 - **1311** - 3320

**P2:** 2130 - 2120 - 1110 - 2210 - 1220 - 3230 - 1310 - 3320 - 2330

Fig. 3c Parents in Fig. 3a converted to generate Of1A.

**P1:** 1110 - 2330 - 2120 - 2130 - 2210 - 1220 - 3230 - **1311** - 3320

**P2:** **2131** - 2120 - 1110 - 2210 - 1220 - 3230 - 1310 - 3320 - 2330

Fig. 3d Parents in Fig. 3a converted to generate Of1B.

**P1:** 1111 - **2330** - 2120 - 2130 - 2210 - **1220** - 3230 - 1311 - 3320

**Of1:** 1111 - **1220** - 2120 - 2130 - 2210 - **2330** - 3230 - 1311 - 3320

Fig. 4a A sample procedure of Mutation 1.

**P1:** 1111 - **2330** - 2120 - 2130 - 2210 - 1220 - 3230 - 1311 - 3320

**Of1:** 1111 - **1330** - 2120 - 2130 - 2210 - 1220 - 3230 - 1311 - 3320

Fig. 4b A sample procedure of Mutation 2.

problem with alternative production routing. The problem formulation discussed in section II will be adopted to model two sample problems originally obtained from Lee and DiCesare [13] and Hoitomt et al. [14].

*A. Example 1*

This model is obtained from Lee and DiCesare [13]. The model consists of 5 jobs, 3 machines, 4 operations in each job, and each operation can be performed on one of the 3 machines as shown in Table I. The lot size for the jobs is 10, and the objective function is to minimize the makespan, equation (1.1). Originally, Lee and DiCesare [13] applied Petri Nets combined with the heuristic search to obtain 439 makespan. Recently, Kumar et al. [15] have applied Ant Colony approach to obtain 420 makespan. By applying the proposed DGs in GA, this paper obtained 360 makespan as shown in Table II.

TABLE I MODEL PARAMETERS OF EXAMPLE 1

Job	Process			Job	Process		
	Oper.	m/c	Time		Oper.	m/c	Time
1	1	1	7	4	1	2	9
	1	3	4		1	3	5
	2	2	3		2	1	6
	3	1	3		2	3	2
2	3	3	6	3	2	7	
	4	1	2	3	3	12	
	4	2	4	4	1	9	
	1	1	8	4	2	6	
3	1	2	12	5	1	1	10
	2	3	4		1	3	15
	3	1	7		2	2	7
	3	2	14		2	3	14
4	4	1	8	3	1	5	
	4	3	4	3	2	8	
	1	1	10	4	1	4	
	1	2	15	4	2	6	
5	1	3	8	4	3	8	
	2	2	2				
	2	3	6				
	3	1	2				
6	3	3	4				
	4	1	6				
	4	2	3				

*B. Example 2*

This model is obtained from Hoitomt et al. [14]. The problem size contains 127 jobs, 33 machines, a maximum of 7 operations in each job, and each operation may perform on one of the 6 suitable machines. The objective function is to minimize a weighted quadratic tardiness function of the jobs, equation (1.2). They applied Lagrangian relaxation to solve this problem. The capacity constraints and precedence constraints were relaxed. The feasible solution obtained has a value of  $Z = 247,843.5$ . This problem has been studied by other researchers. Zhao and Wu [16] applied GA, while Yang and Wu [17] proposed an Adaptive GA to solve the same problem. The results are shown in Table III.

TABLE III RESULTS OF EXAMPLE 2 BY DIFFERENT APPROACHES

	Approach	Z
Hoitomit et al.	LP Relexation	247,843.5
Zhao and Wu	SGA	239,740.0
Yang and Wu	AGA	239,365.0
Proposed DG	DG	239,294.5

**V. CONCLUSION**

From Examples 1, and 2, the results indicate that the proposed DGs approach perform better than the other heuristic approaches (such as, Petri Nets, Ant Colony, Simple GA, and Adaptive GA) in the sample FMS scheduling problems.

**VI. REFERENCES**

- [1] R. Cheung, M. Gen, and Y. Tsujimura, "A tutorial survey of job-shop scheduling problems using genetic algorithms – I", *Computers Industrial Engineering*, vol. 30 no. 4, pp. 983-997, 1996.
- [2] R. Cheng, M. Gen, and Y. Tsujimura, "A tutorial survey of job-shop scheduling problems using genetic algorithms – II", *Computers Industrial Engineering*, vol. 37 no. 1, pp. 51-55, 1999.
- [3] A.K. Jain, and H.A. ElMaraghy, "Single process plan scheduling with genetic algorithm", *Production Planning and Control*, vol. 8 no. 4, pp. 363-376, 1997.
- [4] S. Cavalieri, and P. Gaiardelli, "Hybrid genetic algorithms for a multiple-objective scheduling problem", *Journal of Intelligent Manufacturing*, vol. 9, pp. 361-367, 1998.
- [5] Sakawa, M., (2002), *Genetic algorithms and fuzzy*

TABLE II PRODUCTION SCHEDULING OF EXAMPLE 1

M1	J501	J201	J503	J303	J203	J504																														
M2		J502	J302	J102	J403	J104	J304																													
M3	J301	J401	J101	J402	J202	J103	J404	J204																												
K	10	20	30	40	50	60	70	80	90	100	110	120	130	140	150	160	170	180	190	200	210	220	230	240	250	260	270	280	290	300	310	320	330	340	350	360

- multi-objective optimization*, Kluwer Academic Publisher, pp. 188-222.
- [6] M. Mori, and C.C. Tseng, "Genetic algorithms for multi-mode resource constrained project scheduling problem", *European Journal of Operational Research*, vol. 100 no. 1, pp. 134-141, 1997.
- [7] N. Jawahar, P. Aravindan, and S.G. Ponnambalam, "A genetic algorithm for scheduling flexible manufacturing systems", *International Journal of Advanced Manufacturing Technology*, vol. 14, pp. 588-607, 1998.
- [8] F. Ghedjati, "Genetic algorithms for the job-shop scheduling problem with unrelated parallel constraints: Heuristics mixing method machines and precedence", *Computers and Industrial Engineering*, vol. 73, pp. 39-42, 1999.
- [9] Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs*. Springer-Verlag Berlin Heidelberg New York, 1996.
- [10] J.J. Grefenstette, "Optimization of control parameters for Genetic Algorithms", *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 16 no. 1, pp. 122-128, 1986.
- [11] S. Cavalieri, and P. Gaiardelli, "Hybrid genetic algorithms for a multiple-objective scheduling problem", *Journal of Intelligent Manufacturing*, vol. 9, pp. 361-367, 1998.
- [12] E.L. González, and M.A.R. Fernández, "Genetic optimization of a fuzzy distribution model", *International Journal of Physical Distribution and Logistics Management*, vol. 30 no. 7/8, pp. 681-696, 2000.
- [13] D.Y. Lee and F. DiCesare, "Scheduling Flexible Manufacturing Systems Using Petri Nets and Heuristic Search", *IEEE Transactions on Robotics and Automation*, vol. 10, no. 2, 1994, pp. 123 – 132.
- [14] D.J. Hoitomt, P.B. Luh and K.R. Pattipati, "A Practical Approach to Job-Shop Scheduling Problems", *IEEE Transactions on Robotics and Automation*, vol. 9, no. 1, 1993, pp. 1 – 12.
- [15] R. Kumar, M.K. Tiwari and R. Shankar, "Scheduling of flexible manufacturing systems: an ant colony optimization approach", *IMechE*, vol. 217, Part B, 2003, pp. 1443 – 1453.
- [16] C.W. Zhao and Z.M. Wu, "A Genetic Algorithm Approach to the Scheduling of FMSs with Multiple Routes", *The International Journal of Flexible Manufacturing Systems*, vol. 13, 2001, pp. 71 – 88.
- [17] H.H. Yang and Z.M. Wu, "The application of Adaptive Genetic Algorithms in FMS dynamic rescheduling", *International Journal of Computer Integrated Manufacturing*, vol. 16, no. 6, 2003, pp. 382 – 397.