World Academy of Science, Engineering and Technology
International Journal of Industrial and Manufacturing Engineering
Vol:6, No:4, 2012

A Multi-Level GA Search with Application to the Resource-Constrained Re-Entrant Flow Shop Scheduling Problem

Danping Lin, C.K.M. Lee

***Abstract*—**Re-entrant scheduling is an important search problem with many constraints in the flow shop. In the literature, a number of approaches have been investigated from exact methods to meta-heuristics. This paper presents a genetic algorithm that encodes the problem as multi-level chromosomes to reflect the dependent relationship of the re-entrant possibility and resource consumption. The novel encoding way conserves the intact information of the data and fastens the convergence to the near optimal solutions. To test the effectiveness of the method, it has been applied to the resource-constrained re-entrant flow shop scheduling problem. Computational results show that the proposed GA performs better than the simulated annealing algorithm in the measure of the makespan

***Keywords*—**Resource-constrained, re-entrant, genetic algorithm (GA), multi-level encoding

I. Introduction

RE-ENTRANT flow shop scheduling problem (RFSP) is an important industrial problem that faced by many semiconductor plants, repairing companies or electronic companies. In such assembly lines, the work in process will repeatedly pass through some of the work stations at different stages of operation with the same order. When each machine processes the set of all jobs in the same order too, this problem evolves into the re-entrant permutation flow shop scheduling problem (RPFSP), which is the special case of RFSP. As RFSP is known to be NP-hard, when resource constraints are taken into consideration, the searching space is highly constrained. In some production lines, the production execution mode has direct relation with the re-entrant chance. Take the electro-coating as example, high-end machines such as high pressure sprayer offer high performance and consume more paint and are less prone to rework, whereas skilled worker can make up the resource consumption with more time allocation but with a higher possibility of re-work. However, the welding process is the opposite case. Manual work promises better welding within the tolerance of quality standard, whereas automatic welding costs more filler material and has higher chance of rework. Hence, according to the application, the practical importance and theoretical challenge, there is a strong motivation to explore the scheduling problem of alternative execution mode which is related to the rework probability. This kind of scheduling solution has a higher pay-off when there is limited expensive equipment and high skilled labor. In general, meta-heuristic methods differ from exact methods in the sense that not only the meta-heuristics methods can obtain a nearly optimal solution within a reasonable computational time but also it can manipulate some solutions to attain solutions with higher quality.

Danping Lin is with Nanyang technological university, Singapore e-mail : Lind0020@e.ntu.edu.sg

Therefore, the resource-constrained re-entrant scheduling problem in the flow shop needs such tools. Introduced in [1], genetic algorithm serves as an important meta-heuristic strategy to solve the hard optimization problems. So in this paper a genetic algorithm with multi-level encoding approach is proposed to solve the resource-constrained re-entrant flow shop scheduling problem where the job's re-entrant possibility is depend on the execution mode and resource consumption. The proposed GA will integrate the tasks of job scheduling and execution mode selection, which are subject to the precedence constraints and capacity constraints of machine and human resource, such that the makespan of the schedule is minimized.

The rest of this paper is devoted to describing the problem's characteristic with the multi-level encoding and its application to the practical problem. In section II, the related work is discussed, and in section III the proposed method is presented. Section IV conducts the proposed method with the case study and the results are presented and analyzed. Section V sums up the study and remarks the future directions.

II. Literature Review

A. Related Work of Re-entrant Flow Shop Scheduling

The problem of re-entrant flow shop scheduling can be traced to the work of Graves et al. [2] who is the first one to systematically define the RFSP term as the problems happened in the shop area with a sequence of operations are repeatedly processed by machines.

However this problem was only studied broadly since 2000 with the demanding manufacturing requirement from the semiconductor industries or integrated circuit fabrication or thin film transistor liquid crystal display process. Exact method, heuristic and meta-heuristic methods all have been proposed for solving the re-entrant flow shop scheduling problem. However, the comprehensive analysis of this problem is not done till the review paper of Lin and Lee [3] which classifies the re-entrant problems and analyzes the optimization methods of re-entrant manufacturing scheduling.

Mostly exact methods formulate the re-entrant flow shop scheduling problem as the integer programming problem and simulate the efficiency of the new proposed approaches. However, the computational complexity is high as long run time is used to narrow down the space within the constraints. The referenced work can be found from [4-15].

Heuristic techniques such as dispatching rules and constructive heuristics have also been applied to the re-entrant flow shop scheduling problem. Recent work that applied the dispatching rules in the semiconductor industries are found from [16-18]. They always tested the proposed modes in terms of simulation. Though these heuristic methods produce good quality solutions, they are not extensively used during the period of 2000-2007 as the running time are often large and increase rapidly with the size of the problem increases.

Meta-heuristic, especially genetic algorithm (GA), evolved to be the new trend for the re-entrant flow shop scheduling problem. GA gets the edges over other techniques in the area of large-scale complexity like the very large scale integrated (VLSI) circuit manufacturing or repairing industries which are highly re-entrant, and involve hundreds of machines, restrictions and processing steps. Therefore, [19-23] used GA to solve the re-entrant problem and mainly focus their objectives on minimizing the makespan. Another meta-heuristic method tabu search has also been used, which can be found from [24-25].

Going through the related work, we found that exact methods and meta-heuristics are the main tools based on the frequency of used times and most of them adopted minimizing the makespan as their objectives. However, most of the works assumed unlimited buffer or no resource constraint in the processing procedure, or some of the work relaxed these constraints and decomposed the problem into sub-problems; these approaches isolated the resource constraint from the re-entrant flow shop scheduling. Only the work of Scholz-Reiter et al. [15] that fully takes the resource constraints into account. In their work, machines and labor resources are taken into consideration, priority rules and rule combinations are simulated both for small problem instances and long term system behavior with a time span of ten years. The results revealed that the performance of priority rules depends on the utilization level of the system, thus a proper control system is crucial to select and combine the priority rules.

B. Related Work of Flow Shop with Constrained Resource

Generally the works of resource-constrained flow shop can be classified into two types in terms of objectives. The first type related to allocating and scheduling the limited resource to the flow shop. Daniels et al. [25] discussed the feasibility and benefits of dynamic assignment of partially flexible labor resources, linked skill matrices with self-buffering production line and eventually proved the benefit of complete flexibility can be achieved by a reasonable small amount of carefully selected partial flexibility. Similar work is done by Neubert and Savino [26].

The second type of flow shop with limited resource mainly concerns the jobs scheduling problem, while the resource constraints are only one of their considerations. The example can be found from Janiak [27], whose main objective is to minimize the job overall completion time. In his model, the duration of each operation on certain machines is a liner function of the allocated part of a constrained resource. The proposed algorithm is based on disjunctive graph theory and brand & bound algorithm. However, resource allocation is regarded as one of the concerns for scheduling. The similar work has been done by Cheng and Janiak [28] and Leu and Hwang [29]. Although there are a lot of research has been done on flow shop scheduling problem, according to authors' knowledge, no studies have done connecting the resource constraint with the re-entrant possibility but it is quite common for the flow shop circumstance. Therefore, we will propose a GA-based technique that addresses the re-entrant flow shop scheduling problem with resource constraints. Our GA is designed to perform the combination work of job scheduling, resource satisfaction and re-entrant possibility control. Our approach will offer a novel way of chromosome encoding for this specific problem.

III. Multi-level Encoding Genetic Algorithm

A. Multi-level Chromosome Encoding

There is a strong interdependence between the execution mode selection and the re-work possibility, and it is not clear in which order they should be performed. Therefore, it is inappropriate to separate the schedules with execution mode selection and re-entrant possibility control. Instead they should be considered concurrently as the search space for the combined task is large and discrete. GA is known to work well on such complicated problem and the inherent heuristic character of GA allows the synchronous exploration at the same optimization iteration. That is the main reason we choose GA. Moreover, when the integrated optimization requirement reflecting in the GA's chromosome encoding, information need to be kept intact and undistorted as well as easy to inherit. Compared with one-level chromosome encoding, when the number of the re-entrant job is dynamic, the subsequent length of the chromosome is unfixed. Therefore, there can be information mixing among different types of data when carrying out the crossover or mutation operation. To the opposite, the multi-level encoding reserves the data into different levels according to the data types. So the length of each level is allowed to be different. Furthermore, crossover or mutation is implemented level by level, which essentially stops the chance of data mixing and distortion. At the same time, efficient repair operator is used to revise any infeasible solution. The detail of the information representation is shown in Fig. 1.

Job Index	4		6		3		1		5		2	
Re-entrant Possibility	20%	30%	60%	10%	40%	75%	65%	20%	70%	40%	15%	80%
Resource Consumption – Machine	30	20	25	70	50	15	28	36	40	45	30	56
Resource Consumption – Labor	30	20	30	20	30	20	30	20	30	20	30	20

Mode 0 Mode 1

(a) Before Encoding

Job Index	4	6	3	1	5	2
Re-entrant Possibility	0.20	0.10	0.75	0.65	0.70	0.15
Execution Mode	0	1	1	0	0	0

(b) After Encoding

Fig. 1 Multi-level chromosome encoding

Fig. 1 illustrated how the GA translates the problem into the chromosomes. 6 jobs supposed to be scheduled. Before translation, the data is presented as a vector that contains four level's data: first level is six unique indexes for jobs, and the second level contains twelve percentage numbers which individually represent the re-entrant possibilities of each job under two different execution modes. Following the same

meaning, level 3 and 4 are the representation of the machine and labor consumption under two different execution modes. During the encoding, GA will randomly choose either mode 0 or mode 1 as its execution mode, thus its re-entrant possibility and corresponding resource consumption will be tracked. Assumed the combination of execution modes in Fig. 1(b) fulfills the resource constraints, it is obvious that the job 1, 3, 5 have the rework possibility more than 50 percents, which means all these jobs need to be reworked. The re-entrant jobs will be given other indexes to differentiate with the original job indexes. Of course, no matter for the original jobs or those re-entrant jobs, they should within the overall resource constraints. Therefore, there are actually nine jobs need to be scheduled with the aim of minimizing the total makespan. After that GA can do the crossover and mutation again just for the level of job sequences. It is worth to note that the crossover and mutation operators are executed level by level.

In this meaning, GA will randomly generate an initial population for each generation. During each generation, two parents are chosen to propagate new offspring by applying the crossover and mutation. Later the propagation effect is evaluated by the fitness function. At the same time the function of the repair operator is twofold. Firstly, it makes sure infeasible solutions that violate the resource constraints are amended. It is done by mutating the allele in the execution mode. Secondly, it helps to repair those infeasible solutions that produced during the scheduling procedures where the re-entrant jobs may be placed before the original jobs. It is done by swapping the positions of the re-entrant jobs and original jobs. During this swap, the incidental data of the execution mode and re-entrant possibility is kept constant.

B. Problem Formulation

The objectives of the proposed GA are to (1) find the solutions to the re-entrant flow shop scheduling problem that minimizing the makespan, (2) find out the execution mode of each job as well as satisfy the resource constraints. Set $r_{imk} = 1$ if resource *k* is used for machine *m* for conducting the job *i* and 0 otherwise. Therefore, the fitness function is set the same as the objective function as

$$\text{Minimize } \sum_{i,m=1}^{n+r} p_{im} x_{im} \quad (1)$$

$$\text{S.T.} \quad \sum_{m \in M_i} x_{im} = 1, \forall i \in N \quad (2)$$

$$t_i \geq 0, \forall i \in N \quad (3)$$

$$t_i + p_i \leq t_{i+1,} \forall i \in N \quad (4)$$

$$y_{ijm} = y_{jim}, \forall i, j \in E_m, \forall m \in M_i \quad (5)$$

$$y_{ijm} \leq 0.5(x_{im} + x_{jm}) \leq y_{ijm} + 0.5, \forall i, j \in E_m, \forall m \in M_i \quad (6)$$

$$\sum_{j \in E_m} Z_{ijm} \leq 1, \forall i, j \in E_m, \forall m \in M_i \quad (7)$$

$$\sum_{j \in E_m} Z_{jim} \leq 1, \forall i, j \in E_m, \forall m \in M_i \quad (8)$$

$$z_{ijm} + z_{jim} \leq 1, \forall i, j \in E_m, \forall m \in M_i \quad (9)$$

$$t_i \leq t_j, \forall i, j \in B \quad (10)$$

$$\sum_{m \in M_i} \sum_{i \in N} r_{imk} x_{im} \leq R_k, \forall i \in N, , \forall m \in M_i \quad (11)$$

$$x_{im}, y_{ijm}, z_{ijm}, r_{imk} = 0 \text{ or } 1, \forall i, j \in N, \forall m \in M_i \quad (12)$$

Where the problem parameters are:

N: the set of all jobs

n: the number of jobs

i, j: job index

r: re-entrant job index

m: machine index

M_i: the set of machines to process job *i*

E_m: the set of jobs that might be processed on machine *m*

B: the set of pairs of jobs between which there is precedence relationship, when job *i* must precedence job *j*

p_{im}: processing time of job *i* in machine m

Decision variables:

$x_{im} = 1$, if operation of job *i* is assigned to machine m; 0, otherwise;

$y_{ijm} = 1$, if operation of job *i* and *j* are assigned to the same machine *m*; 0, otherwise;

$z_{ijm} = 1$, if operation of job *i* immediately precedes *j* on machine *m*;

$r_{imk} = 1$, if operation of job *i* is conducted on machine *m* with k resource consumption;

t_i is the starting time of job *i* ;

C_{max} is the completion time of the last job.

Eq. (1) minimizes the total makespan of jobs as a main objective. Constraints (2) ensure that each operation must be processed by exactly one machine. Constraints (3) ensure that each operation begins after time zero. Constraints (4) ensure that the order of operation of each job is respected. Constraints (5) and (6) ensure that $y_{ijm} = y_{jim} = 1$ when $x_{ijm} = x_{jim} = 1$. Constraints (7) and (8) ensure that each operation has at most one predecessor and successor on machine *m*. Constraints (9) ensure that z_{ijm} and z_{jim} cannot equal to 1 simultaneously. Constraints (10) determine the set of pairs of jobs between which there is precedence relationship. Constraints (11) promise the resource constraints are respected. Constraints (12) are binary constraints.

IV. Case Study and Experimental Results

In order to show the effectiveness of the proposed method, the experiment is based on real case and compared with the homogeneous algorithm of simulated annealing algorithm.

EG (Alias) Ltd is one of the magnates of providing various electronic technology and services in broad domain. In south-east Asia, it endeavors in offering a wide range of repairing service to huge energy parts. Their repairing workflow is the typical flow shop as the parts will go through receiving inspection, disassembly, repairing procedure and final assembly following the same order. Before parts moving on to the new workstations, they will be inspected. When new faults are found they need to return to the previous workstations to be reworked. The rework possibility is not constant, because

the execution modes of the previous workstations will influence the consumption of machine time and worker time, and the re-entrant chance. Take the power nozzle repairing procedure as example, if the job's due date is not so tardy, the workstation may choose the more complicated repairing mode that will go through 40 steps. Under this mode, a typical piece of power nozzle will cost 400 units of worker time, 120 units of machine time with 9 percents of rework possibility. If the job is emergent, the same piece of part can be done applying the compact mode that only goes through 13 steps, consuming 215 units of worker time, 420 units of machine time with as high as 15 percent of chance to be reworked. Whatever the original jobs or the re-entrant jobs, the differences of the execution modes for them are the processing time in some particular stages and the consumption of the machine and manpower resources and some of the parts do not need to pass all the machines.

So the proposed multi-level GA is simulated to minimize the makespan for this plant. The experiments are conducted into 20-, 30-, and 40-job sizes according to the plant's current job classification. The experiments are implemented in the MATLAB 7.0 on a Pentium 4 3.17GHz PC with 3.25G MB memory. The parameter setting is tuned based on the foregone experiments. The size of the GA population was set as 200, the two-point crossover with crossover probability is set as 0.8, and the Gaussian mutation with mutation probability is set as 0.4. Each of the GA runs is stopped after 100 generations. While for the SA, 10 runs are conducted with fast annealing function and stopped after 200 generations. On the other side, based on the real data's distribution, the processing time in the six workstations is exponential distributed with the mean of 15 and the re-entrant probability is uniformly distributed.

TABLE I
AVERAGE PERFORMANCE OF GA AND SA

Job size	Data set	GA			SA		
		Mean	St. deviation	Best	Mean	St. deviation	Best
20 jobs	Set 1	958.0648	25.71703	**906.969**	971.5597	23.27611	940.78
	Set 2	990.4381	27.3935	**933.85**	994.3035	26.42774	946.12
	Set 3	1015.614	30.8407	**957.22**	1034.202	23.09262	993.30
	Set 4	1157.2	28.31164	**1104.36**	1158.831	25.74849	1116.4
30 jobs	Set 1	1588.413	23.4718	1548.97	1606.738	26.82226	**1548.269**
	Set 2	1451.745	26.707	**1394.879**	1462.596	32.23332	1403.929
	Set 3	1580.758	34.04155	1518.86	1598.082	46.03813	**1513.42**
	Set 4	1602.434	43.05558	**1517.26**	1631.362	42.59203	1555.04
40 jobs	Set 1	1988.06	40.62337	**1910.7**	2021.577	43.99321	1932.029
	Set 2	2202.546	57.62981	**2100.459**	2224.944	77.26172	2103.00
	Set 3	2004.31	38.92844	**1932.4**	2044.212	99.75263	1933.769
	Set 4	1900.255	30.94322	**1857.5**	1953.697	41.13965	1862.429

In the simulation tests, we give the mean, standard deviation and the best performance referring different problem size respectively. Table I shows the solutions found by GA and SA in these ten runs under the resource and re-entrant constraints. Ideally, the GA-based method should converge to the optimal or near-optimal solution in each run. Computing the average of the best solutions found over the ten independent runs provides an indication of the robustness of the search strategy. In ten out of the twelve tests, GA works better than the SA. For the two out of twelve tests showing GA is inferior to SA, the differences of the fitness between them are less than 0.3%. This illustrates that the proposed GA approach is able to consistently converge to high-quality solutions in every GA run for different problem sizes.

V. CONCLUSION

This paper presented a multi-level encoding genetic algorithm for re-entrant flow shop scheduling problem with resource constraints. In the proposed approach, the multi-level encoding translates the data into different levels according to data types and executes crossover and mutation individually for each level. The proposed method has demonstrated better performance with regard to the makespan when compared with simulated annealing algorithm.

The superiority of the multi-level chromosome encoding lies in:

1) It provides integrity and flexibility to perform the combination of searching task.
2) Multi-level encoding enables user to store more information about the problem and this encoding is more clear and visualized.
3) As the re-entrant job number is dynamic, different levels have their own operators. On one hand, it helps to keep the information independently and integrated. On the other hand, it expedites the chromosome converge to the better optimal as one group of multi-level chromosome represents one solution, it does not need to decode.

Referring to the result differences of GA and SA, they may come from the inherited characteristic of the algorithms themselves. SA traverses the search space by applying local transformations to the solution representation, while GA's search ability is decided by the chromosome representation and search operators. In the first place, the multi-level encoding is problem-specific encoding, which stores the information according to the data types so that the crossover and mutation operators can be implemented level by level without distorting and mixing different types of data. Second, when decoding the chromosomes, it is vivid and easy as the one group of multi-level chromosomes directly represents one solution.

Future research may include applying the method to larger problem sizes and broadening the application into semiconductor industries. Improvements to the method could be explored, such as the parameters tuning or the sensitive analysis.

REFERENCES

[1] J. Holland, *Adaptation in natural and artificial systems. .* Michigan: Ann Arbor MI: University of Michigan Press, 1975.
[2] S. Graves, *et al.*, "Scheduling of re-entrant flow shops," *Journal of Operations Management,* vol. 3, pp. 197-207, 1983.
[3] D. Lin and C. K. M. Lee, "A review of the research methodology for the re-entrant scheduling problem," *International Journal of Production Research,* vol. 49, pp. 2221-2242, 15 Apr, 2011 2010.
[4] E. Demirkol and R. Uzsoy, "Decomposition methods for re-entrant flow shops with sequence-dependent setup times," *Journal of Scheduling,* vol. 3, pp. 155-77, 2000.
[5] Y. Park, *et al.*, "Performance analysis of re-entrant flow shop with single-job and batch machines using mean value analysis," *Production Planning and Control,* vol. 11, pp. 537-546, 2000.
[6] N. G. Odrey, *et al.*, "Generalized Petri net modeling approach for the control of re-entrant flow semiconductor wafer fabrication," *Robotics and Computer-Integrated Manufacturing,* vol. 17, pp. 5-11, 2001.
[7] J. H. Pan and J. S. Chen, "Minimizing makespan in re-entrant permutation flow-shops," *Journal of the Operational Research Society,* vol. 54, pp. 642-53, 2003.
[8] E. H. Nielsen, "How does variability in input load relate to the probability of critically delayed delivery in a simple multipart re-entrant flow-line problem?," *International Journal of Production Research,* vol. 42, pp. 3383-3396, 2004.
[9] J.-S. Chen and J. C.-H. Pan, "Integer programming models for the re-entrant shop scheduling problems," *Engineering Optimization,* vol. 38, pp. 577-592, July 2006 2006.
[10] H. Haitham and W. Ruml, "Network flow modeling for flexible manufacturing systems with re-entrant lines," presented at the Proceedings of the 45th IEEE Conference on Decision and Control Piscataway, NJ, USA, 2006.
[11] S. W. Choi and Y. D. Kim, "Minimizing makespan on a two-machine re-entrant flowshop," *Journal of the Operational Research Society,* vol. 58, pp. 572-81, 2007.
[12] S. Jiang and L. Tang, "Lagrangian relaxation algorithms for re-entrant hybrid flowshop scheduling," presented at the 2008 International Conference on Information Management, Innovation Management and Industrial Engineering, Piscataway, NJ, USA, 2008.
[13] D. Yang, *et al.*, "Multi-family scheduling in a two-machine reentrant flow shop with setups," *European Journal of Operational Research,* vol. 187, pp. 1160-1170, 2008.
[14] T. Kaihara, *et al.*, "Proactive maintenance scheduling in a re-entrant flow shop using Lagrangian decomposition coordination method," *CIRP Annals - Manufacturing Technology,* vol. 59, pp. 453-456, 2010.
[15] B. Scholz-Reiter, *et al.*, "Analysis of priority rule-based scheduling in dual-resource-constrained shop-floor scenarios," presented at the International Conference on Advances in Machine Learning and Systems Engineering, Berkeley, CA, United states, 2010.
[16] J. R. Perkins and P. R. Kumar, "Optimal control of pull manufacturing systems," *IEEE Transactions on Automatic Control,* vol. 40, pp. 2040-51, 1995.
[17] R. Cigolini, *et al.*, "Implementing new dispatching rules at SGS-Thomson Microelectronics," *Production Planning and Control,* vol. 10, pp. 97-106, 1999.
[18] I. A. El-Khouly, *et al.*, "Modelling and simulation of re-entrant flow shop scheduling: An applicationin semiconductor manufacturing," presented at the 2009 International Conference on Computers and Industrial Engineering, CIE 2009, Troyes, France, 2009.
[19] H. Hwang and J. U. Sun, "Production sequencing problem with re-entrant work flows and sequence dependent setup times," *International Journal of Production Research,* vol. 36, pp. 2435-2450, 1 September 1998 1998.
[20] J.-S. Chen, *et al.*, "A hybrid genetic algorithm for the re-entrant flow-shop scheduling problem," *Expert Systems with Applications,* vol. 34, pp. 570-577, 2008.
[21] H. Rau and K.-H. Cho, "Genetic algorithm modeling for the inspection allocation in reentrant production systems," *Expert Systems with Applications,* vol. 36, pp. 11287-11295, 2009.
[22] C.-H. Liu, "A genetic algorithm based approach for scheduling of jobs containing multiple orders in a three-machine flowshop," *International Journal of Production Research,* vol. 48, pp. 4379-96, 2010.
[23] C. K. M. Lee and D. Lin, "Hybrid genetic algorithm for bi-objective flow shop scheduling problems with re-entrant jobs," presented at the IEEE International Conference on Industrial Engineering and Engineering Management, IEEM2010, Macao, China, 2010.
[24] M. Nawaz, *et al.*, "A heuristic algorithm for the m-machine, n-job flow-shop sequencing problem," *Omega,* vol. 11, pp. 91-95, 1983.
[25] R. L. Daniels, *et al.*, "Flow shop scheduling with partial resource flexibility," *Management Science,* vol. 50, pp. 658-669, 2004.
[26] G. Neubert and M. M. Savino, "Flow shop operator scheduling through constraint satisfaction and constraint optimisation techniques," *International Journal of Productivity and Quality Management,* vol. 4, pp. 549-68, 2009.
[27] A. Janiak, "General flow-shop scheduling with resource constraints," *International Journal of Production Research,* vol. 26, pp. 1089-1103, 1988.
[28] T. C. E. Cheng and A. Janiak, "A permutation flow-shop scheduling problem with convex models of operation processing times," *Annals of Operations Research,* vol. 96, pp. 39-60, 2000.
[29] S.-S. Leu and S.-T. Hwang, "GA-based resource-constrained flow-shop scheduling model for mixed precast production," *Automation in Construction,* vol. 11, pp. 439-52, 2002.
[30] L.-Y. Tseng and S.-C. Chen, "Two-phase genetic local search algorithm for the multimode resource-constrained project scheduling problem," *IEEE Transactions on Evolutionary Computation,* vol. 13, pp. 848-857, 2009.
[31] H. R. Topcuoglu, *et al.*, "Solving the register allocation problem for embedded systems using a hybrid evolutionary algorithm," *IEEE Transactions on Evolutionary Computation,* vol. 11, pp. 620-634, 2007.
[32] R. Zamani, "An Accelerating Two-Layer Anchor Search With Application to the Resource-Constrained Project Scheduling Problem," *Evolutionary Computation, IEEE Transactions on,* vol. 14, pp. 975-984, 2010.
[33] R. Kolisch and A. Sprecher, "PSPLIB-a project scheduling problem library," *European Journal of Operational Research,* vol. 96, pp. 205-16, 1997.