World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:8, No:5, 2014

Applying Sequential Pattern Mining to Generate Block for Scheduling Problems

Meng-Hui Chen, Chen-Yu Kao, Chia-Yu Hsu, Pei-Chann Chang

Abstract**—**The main idea in this paper is using sequential pattern mining to find the information which is helpful for finding high performance solutions. By combining this information, it is defined as blocks. Using the blocks to generate artificial chromosomes (ACs) could improve the structure of solutions. Estimation of Distribution Algorithms (EDAs) is adapted to solve the combinatorial problems. Nevertheless many of these approaches are advantageous for this application, but only some of them are used to enhance the efficiency of application. Generating ACs uses patterns and EDAs could increase the diversity. According to the experimental result, the algorithm which we proposed has a better performance to solve the permutation flow-shop problems.

Keywords**—**Combinatorial problems, Sequential Pattern Mining, Estimation of Distribution Algorithms, Artificial Chromosomes.

I. Introduction

TO find databases' patterns are very important operation of data mining. For example, like association rule [1] and sequential pattern mining [2]. Scheduling Problems are kinds of the combinatorial problems. According to the definition of the combinatorial problems, it's belonging to NP-hard problem. To find the solution of an optimization problem is a common challenge in which an algorithm may be trapped in the local optima of the objective function when the complexity is high, and there are several local optima in solution space. Tsai et al. [3] and Chun et al. [4] have proposed the algorithms for global optimization problems. The importance of these methods is in many different areas such as modern engineering design and systems operation. Holland [5] and Goldberg [6] proposed Genetic Algorithm (GA) as a tool based on biological mechanisms and natural selection theory. GA has gained much attention regarding its potential as an optimization technique for combinational optimization problems and has been successfully applied in many different areas.

In our earlier research, ACGA had been combined with EDAs using crossover and mutation operators to improve the performance [7]. Main characteristic of ACGA is EDAs alternate with GAs in the evolutionary progress. This research showed that the EDAs can improve solution quality with genetic operators. Some research used other heuristic to solve PFSP, like ant colony optimization (ACO) [8], variable neighborhood search (VNS) [9], local search [10], [11] and artificial bee colony (ABC) [12].

In order to obtain the better performance, some research attempts to find some useful fragment. Use fragment to generate Artificial Chromosome (AC) with good fitness [13].Here we called fragment as block. In previous research [14], [15], we often build the probability model to generate blocks. The blocks focus on the better information in each position. So in this paper, the main idea discusses the relation between jobs and jobs.

Chia-Yu Hsu is with the Department of Information Management, Yuan Ze University, Taoyuan 32026, Taiwan, R.O.C. (Corresponding Author; e-mail: cyhsu@saturn.yzu.edu.tw).

Pei-Chann Chang, Chen-Yu Kao, and Meng-Hui Chen are with the Department of Information Management, Yuan Ze University, Taoyuan 32026, Taiwan, R. O. C.

II. Literature Review

A. Association Rule and Sequential Pattern Mining

Association rule is important in data mining. Association rule is to identify the most important relation-ships and it is generated by analyzing historical data and using the criteria support and confidence. Support shows that how frequently the items appear in the database. The confidence is stood for the judgment which let the rule become useful.

Agrawal et al. [16] discover "what items are bought together in a transaction" over basket data was introduced. The problem is to find what items are bought together from a unprocessed dataset of items is considered that finding intra-transaction patterns. Sequential pattern mining shows the data represented as sequences. Sequential patterns mean that the correlation between transactions [17]. The association rule doesn't consider the order of items, and it focuses on the correlation of each items. The simple step of association rule is shown in Fig 1.

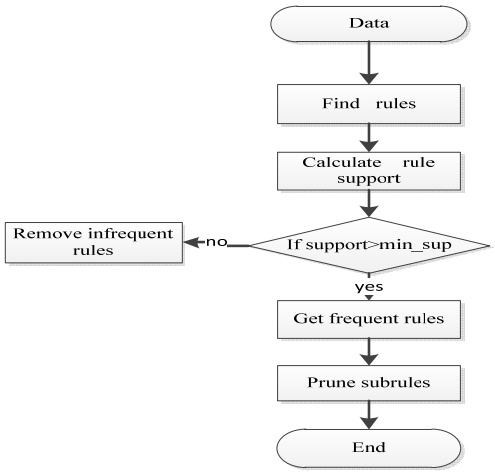

Fig. 1 The flowchart of association rule

B. Estimation of Distribution Algorithms

In EDAs, the problem specific interactions among the

variables of individuals are taken into consideration. In Evolutionary Computations the interactions are kept implicitly in mind whereas in EDAs the interrelations are expressed explicitly through the joint probability distribution associated with the individuals of variables selected at each generation. The probability distribution is calculated from a database of selected individuals of previous generation. Then sampling of this probability distribution generates an offspring. Neither crossover nor mutation has been applied in EDAs, but the estimation of the joint probability distribution associated with the database containing the selected individuals is not an easy task. The flowchart of EDA is shown in the Fig. 2.

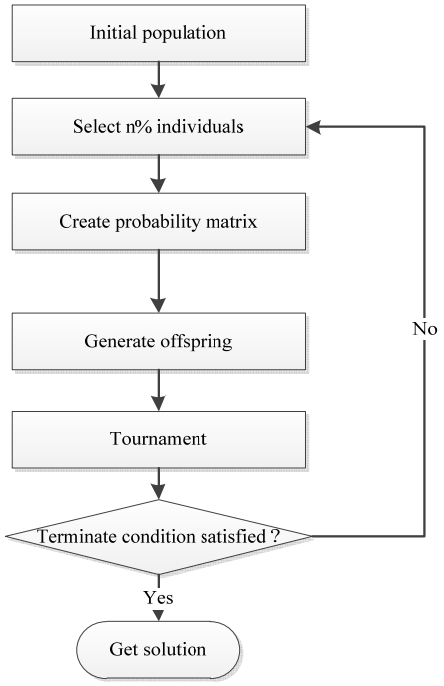

Fig. 2 EDA flowchart

In recently research, some researcher hybrid EDA with other Meta-heuristic algorithms that form a powerful algorithm. In [18], they proposed a new probability model of EDA; capture the variable linkages together with the univariate probabilistic model. A new concept of EDA instead of the most EDAs could use only one statistic information. They combine the new concept of EDA with GA, and join Variable Neighborhood Search (VNS) approach. Their algorithms able get good solution on flow-shop problem.

III. Sequential Pattern Mining to Generate Block

The purposed method in this paper is based on sequential pattern mining and association rule. We proposed an efficient algorithm to solve the combinatorial optimization problems. We called block based on sequential pattern mining as BBSPM. The first stage is getting the excellent solutions from population and store the information from excellent solutions to dominance matrix. The dominance matrix will show the correlation between jobs and jobs. According to the correlation, we could generate blocks by the criteria support and confidence. Then we use the blocks to generate artificial chromosomes to improve the structure of the solutions. The artificial chromosomes are not always generated by full of blocks, it may combine the blocks and other items. So the other items will be according to the estimation of distribution algorithms to find the strength of each item in each position. The competition between the blocks to make sure to maintain high quality of the blocks in achieves. We developed two types of different combinations mechanism of artificial chromosome. Using these blocks we combination artificial chromosome (AC) with higher advantages which speed up the evolution and convergence. In third stage, we modify the Edge histogram based on sampling algorithm with template (EHBSA) mechanism form EDA [19]. We proposed a rapid convergence approach called modified EHBSA to execute in the later iterations. Modify EHBSA can speed up the convergence and these solutions are close to optimal solution.

The flowchart of BBSPM is represented in Fig. 3. The description of each step is as follow:

Step1. Initial Population

To generate initial solutions randomly.

Step2. Calculate Fitness

Calculate fitness for each solution. In this paper, fitness of the experimental problem is Cmax.

Step3. Update the Dominance Matrix

Collecting the better solutions with higher fitness is to update the dominance matrix. In order to avoiding bad information keep in dominance matrix. We make a mechanism to reset the dominance matrix.

Step4. Using Sequential Pattern Mining to Generate Block

In this step, we set a support to define an useful information and the support means the frequency of each job appeared in each position. If the support was larger than threshold which we set, it would be a candidate block.

Step5. Generate artificial chromosomes

There are two methods to generate artificial chromosomes in this paper. We design two mechanisms to generate artificial chromosomes in order to increasing the diversity and keep the higher quality of fitness.

Step6. Using a Local Search Method to Make the Solutions Evolved and Choose Better Solutions to Next Iterations

We improve the EHBSA of EDA to increase ability of search solutions. Then using tournament selection to choose two solutions and competition between each other, better solution can go to next iteration. Maintain the solutions in population with higher diversity.

The above Step 2 to 6 repeats until meet termination criterion set by user.

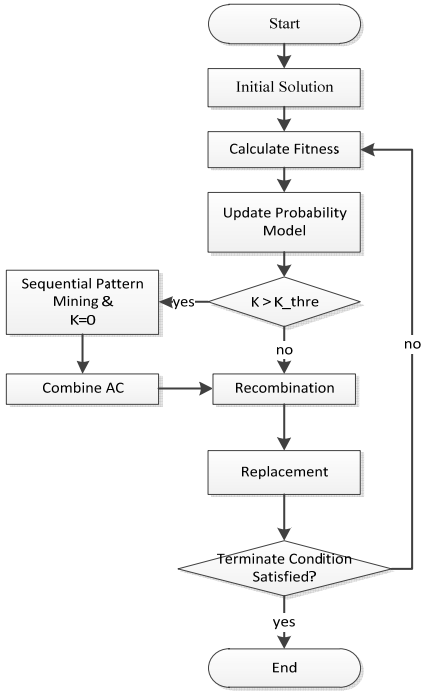

Fig. 3 The architecture of BBSPM

A. Dominance Matrix and Dependency Matrix

Dominance matrix is able to preserve information of high-fitness solutions. Accumulative these jobs positions in each solutions to dominance matrix. Provide useful information for mining blocks and generate AC. In the beginning, calculate fitness for solutions and sort (ascending). Select top N% solutions of population with better fitness value, and store each jobs position in each solution into dominance matrix. In this research, the number of the initial solutions is assigned as 100. We calculate the fitness of all solutions in population; and we select top 30 solutions with higher fitness. Jobs position in each solution is accumulate to dominance matrix. The composition of the matrix is updated by the accumulation of the probability to maintain the competitive approach. The probability matrix is called Accumulative Probability Matrix which is represented as Fig. 4.

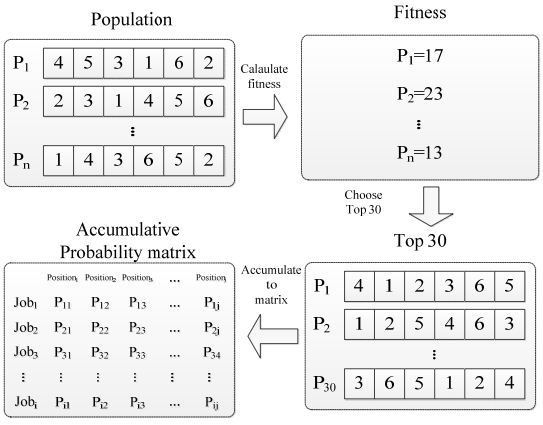

Fig. 4 Accumulative dominance matrix

Accumulate detail is shown in Fig. 4.

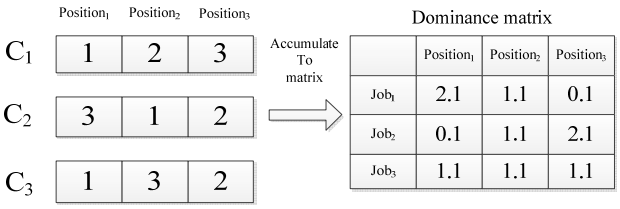

Fig. 5 Accumulate detail of dominance matrix

As Fig. 5, the support of the job1 in position 1 is two times. So by store the correlation between jobs and positions of the solutions which with higher fitness in dominance could shows the strength of each job in each position and the support of each job in each position.

B. Block and Artificial Chromosome Mining Procedure

Each solution has different information in its structure. So if we collect the information of the high-fitness solutions, it would be useful to find high-performance solutions. In this paper, we collect the top n% higher fitness solutions to dominance matrix, and the dominance matrix will show that the times of the each jobs appeared in each position. When the times are bigger, it means that the job has strength in this position. Blocks are design to keep the better information, and combine the blocks to generate artificial chromosomes. Using blocks applied for solving combinational problems has two reasons. One is decreasing the complexity of problems. As shown in Fig. 6, if we could find some information to combine a set of jobs to be grouped together. For example, job 1 with job 8, job 4 with job7, and job10 with job 5, and the total number of feasible solutions will be only 7!. If a longer block is defined, the total number of feasible solutions will be reduced even more abruptly. However, the quality of the blocks will be the key to the block based approach. The other would enhance evolution effect and accelerate the speed of convergence. Because the blocks are combined the better information, it means that the solutions' structure include blocks will cause the schedule order is approach to the order of the optimal solution.

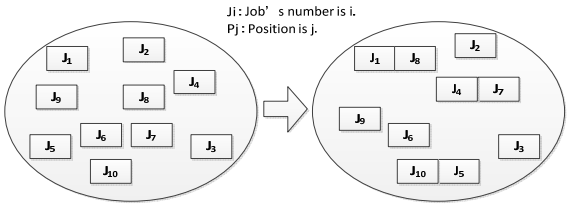

Fig. 6 Complexity reduction

In this paper, the mechanism of generating blocks use sequential pattern mining and dominance matrix. At first, we randomly choose a job within strength in the position according to the dominance matrix, and randomly select two solutions from the top n% fitness solutions. Then find all the support of combining jobs and jobs is higher than the threshold. These all combinations will be the candidate blocks. After weeding out the combinations with the same jobs or positions, the others are the blocks.

For example as Fig. 7 and the threshold is set to 2, defined the block max size is *l*, randomly choose the start position of block is position 1.To calculate the support of job1 to job5 which located position 1, the support of job 1 satisfy the threshold. So Let job 1 is placed in the first position of the block. Then randomly select two useful 1-length job from matrix and combine 2-length job to compare if support bigger than threshold or not from the top n% fitness solutions. Following the same way, finding all the combination supports of each jobs, and to weed out the repeat jobs or positions. The others are as blocks and the block size will be 2 to *l*.

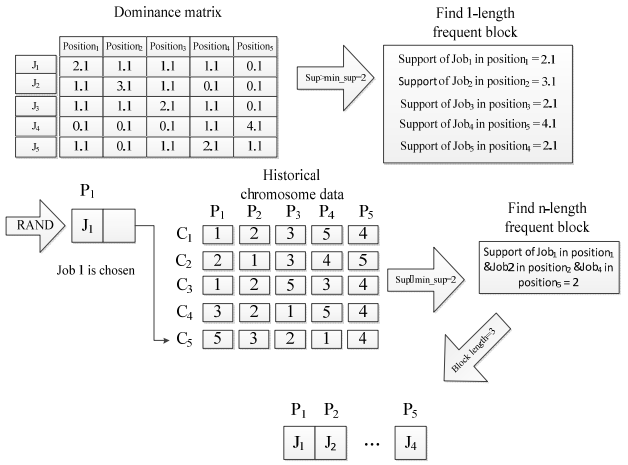

Fig. 7 Block mining procedure

As mentioned above, we have discussed how to combine the blocks. Then this section will show about what's the timing to generate blocks. In fact, the dominance matrix collects the information form solutions. If the structure of the dominance matrix does not change a lot, the blocks also could not change a lot. If we mined the block for each iteration, it would waste more times, and it is not helpful to evolving. So we set an AC counter to accumulate the number of generations to be performed. When AC counter is larger than threshold *A* then execute mining block and generate AC procedure. A block with size K can be generated according to the following procedures: 1. A position of job is picked randomly. 2. According to the dominance matrix, calculate probability of Job1 to Job n in this position. Choose one job and put into this position by roulette wheel selection. 3. Combine 1 to *l* length candidate block and contrast threshold, repeat the step until the block length to achieve *l*. Calculate generated block fitness and save to block archive. 4. If any block with a city overlaps with the blocks previous generated, compare Cmax pare block. The inferior block will be abandoned. 5. The procedures will be repeated again until a pre-defined number of blocks are satisfied.

In this paper, we design two mechanisms of combinations AC. The first one we called AC-I, AC-I and applied in top 75% of generations. The concept of AC-I is shown in Fig. 8, job 2 has been choose into position1 by roulette wheel selection. Check achieve block that whether any blocks' start position has position1 and job number is 2. If achieve then all elements in block will copy to AC. Otherwise, just insert job to AC which selected by roulette wheel selection. This is not mandatory to inject all blocks into AC; the only purpose is increase diversity of AC.

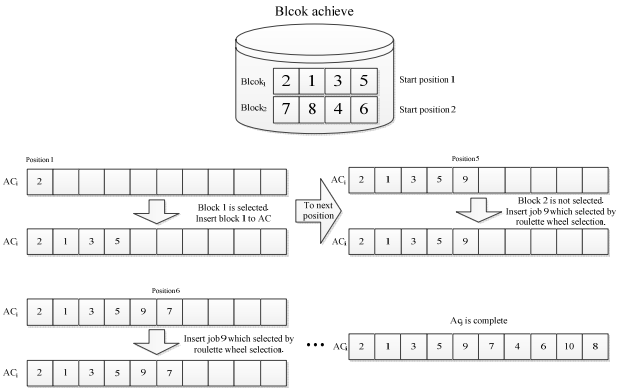

Fig. 8 Artificial chromosome mechanism-I

Another mechanism of combination AC is AC-II, AC-II executedinlast25% of the number of generations. The concept of AC-II is shown in Fig. 9, all blocks in block achieve will inject to AC at the first time. The remaining jobs which select by roulette wheel selection and insert to the remaining space of AC. At last generation, these blocks in block achieve already have good effect and stability of quality. Inject blocks into AC at first time, can make sure quality of AC and speed up the convergence of AC.

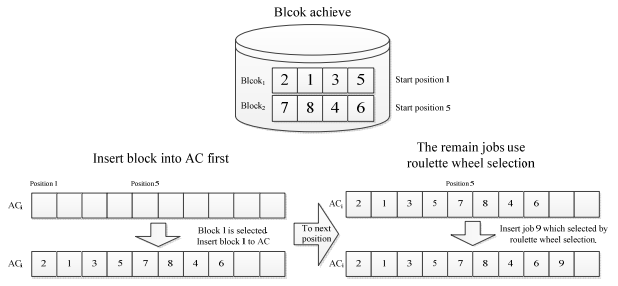

Fig. 9 Artificial chromosome mechanism-II

AC-I and AC-II both use roulette wheel selection to select jobs by these probabilities, and each job's probability is according to the combination probability which merge by each probability of dominance and dependency matrix. The combination probability (CP_{jobn}) defined as follow:

$$CP_{jobn}=(W_{dom}*P_{jobn}^{dom})+(W_{dep}+P_{jobn}^{dep}) \quad (1)$$

The value of weight of dominance matrix (W_{dom}) and weight of dependency matrix (W_{dep}) are depending on user defined. In this research, we defined the value of $W_{de\,p}$ is between 0.3 and 0.7. W_{dep} is increasing as generation goes on. Number of W_{dep} is 1 minus W_{dep}.

C. Modified Ehbsa

EHBSA is an important evolving process of EDA,

EHBSA/WT is using a template in sampling a new string. In generating each new individual, a template individual is chosen from P(t) (normally, randomly). The n (n>1) cut points are applied to the template randomly. When n cut points are obtained for the template, the template should be divided into n segments. Then, we choose one segment randomly and sample nodes for the segment. Nodes in other n-1 segments remain unchanged. We denote this sampling method by EHBSA/WT/n. Since average length of one segment is L/n, EHBSA/WT/n generates new strings which are different at most L/n nodes on average from their templates.

In this paper, we propose a different rule instead of the origin EHBSA/ WT. As Fig. 10, first step is to randomly select a fragment from chromosome. The second step is to switch each gene in this fragment, and to calculate the fitness in each exchange. Finally selecting the exchange with the best fitness is as a new AC.

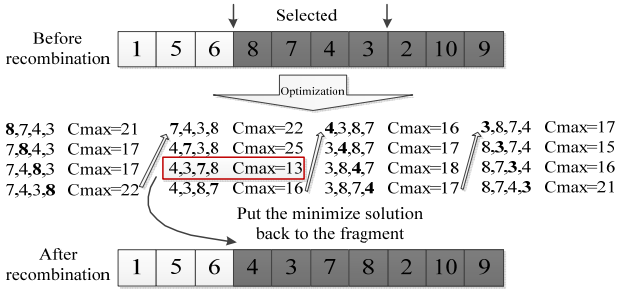

Fig. 10 Modify EHBSA procedure

TABLE I
PERFORMANCE COMPARISON OF TAILLARD'S INSTANCE

			PREDA	JEDA	SGGA	ESGGA	BBSPM
Instance	n,m	Opt.	Error Rate	Error Rate	Error Rate	Error Rate	Error Rate
Ta001	20,5	1278	1.20%	1.16%	1.20%	1.16%	0%
Ta002	20,5	1359	0.88%	0.49%	0.46%	0.36%	0.18%
Ta003	20,5	1081	2.04%	1.73%	1.19%	0.50%	0.04%
Ta004	20,5	1293	0.78%	0.70%	0.90%	0.48%	0.29%
Ta005	20,5	1235	0.82%	1.21%	0.92%	0.70%	0.15%
Ta006	20,5	1195	1.06%	1.54%	1.36%	1.00%	0.26%
Ta011	20,10	1582	0.95%	0.56%	0.82%	0.71%	0.15%
Ta012	20,10	1659	1.24%	1.10%	0.53%	0.28%	0.86%
Ta013	20,10	1496	1.53%	1.50%	1.18%	0.82%	0.76%
Ta014	20,10	1377	1.31%	1.15%	0.77%	0.87%	0.31%
Ta015	20,10	1419	0.98%	0.93%	0.84%	0.66%	0.67%
Ta016	20,10	1397	0.87%	1.50%	0.90%	0.61%	0.58%
Ta041	50,10	2991	3.57%	3.61%	2.95%	3.00%	2.32%
Ta042	50,10	2867	2.34%	3.30%	3.16%	2.75%	1.7%
Ta043	50,10	2839	3.38%	3.90%	3.13%	2.74%	2.37%
Ta044	50,10	3063	1.30%	1.98%	1.40%	1.05%	0.36%
Ta045	50,10	2976	3.09%	3.84%	3.29%	2.15%	1.69%
Ta046	50,10	3006	2.36%	3.00%	2.75%	2.35%	1.03%
Ta081	100,20	6202	5.53%	5.85%	5.96%	5.46%	3.83%
Ta082	100,20	6183	4.02%	4.05%	4.02%	3.37%	3.71%
Ta083	100,20	6271	3.77%	3.91%	3.81%	3.29%	3.27%
Ta084	100,20	6269	2.95%	3.67%	3.44%	3.19%	2.53%
Ta085	100,20	6314	3.97%	4.46%	4.40%	4.05%	3.33%
Ta086	100,20	6364	4.27%	4.58%	4.41%	3.99%	3.28%
Avg.			2.26%	2.49%	2.24%	1.90%	1.40%

IV. EXPERIMENTAL RESULTS

In this section we present the experimental results of the BBSPM and compare the performance of BBSPM with other algorithms. This research adopted the instances of Reeves and Taillard in OR-Library to validate the performance. Executed 30 runs for each instances, n represents the job number and m represents the machine number. Table I shows the performance comparison on Taillard instances. The comparison standard is based on Chen et al. [20]. From the result of BBSPM, the average error rate is 1.4%, outperforms the other algorithms. From the result in Table I, BBSPM has good performance better than the other algorithms.

V. CONCLUSION

In this research, we proposed an algorithm called BBSPM combine sequential pattern mining and EDA. Using sequential pattern mining to generate blocks and the artificial chromosomes are combined by blocks and EDA rule. By the information from two different methods, the artificial chromosomes could get more diversified and effective information. Besides the information from the two different sources, we also design two types of block-based AC generation approaches which are the better combination and competitive rule for artificial chromosomes. Finally, by the modified EHBSA mechanism from EDA, it would be more

powerful convergence approach. From the experimental result validated the idea of application of the AC injection and use modified EHBSA can help the evolutional algorithm to enhance the searching ability. Furthermore, BBSPM has good performance to escape the local optima in the large instances. The research at the present stage is focus on the permutation flows-hop problems, and we will prepare to try other kinds of combinational problems in future to prove the robust of BBSPM.

ACKNOWLEDGEMENT

This research was supported by National Science Council, Taiwan (NSC 102-2221-E-155-093).

REFERENCES

[1] R.Agrawal, T.Imielinski and A.Swami, 1993. MiningAssociation Rules between Sets of Items in Large Databases.In Proc. of the 1993 ACM-SIGMOD Conf. on Management ofData, 207-216.
[2] Agrawal, R. and Srikant, R. 1995. Mining Sequential Patterns, In *Proc. of the 11th Int'l Conf. on Data Engineering*, 3-14.
[3] J. T. Tsai, W. H. Ho, T. K. Liu, and J. H. Chou, "Improved immune algorithm for global numerical optimization and job-shop scheduling problems," Applied Mathematics and Computation, Vol. 194, pp. 406-424, Dec. 2007.
[4] J. S. Chun, H. K. Jung and S. Y. Hahn, "A Study on Comparison of Optimization Performances between Immune Algorithm and other Heuristic Algorithms," IEEE Transactions on Magnetics, vol. 34, No. 5, Sept. 1998.
[5] J. H. Holland, "Genetic Algorithms and the Optimal Allocation of Trials," SIAM J. Comput, vol. 2, pp. 88-105, 1973.
[6] D. E. Goldberg, "Genetic Algorithms in Search, Optimization and Machine Learning (Book style)," Boston, MA: Addison-Wesley, 1989.
[7] P. C. Chang, S. H. Chen, & Fan, C. Y. (2008). "Mining gene structures to inject artificial chromosomes forgenetic algorithm in single machine scheduling problems," Applied Soft Computing Journal, 8(1), 767–777.
[8] Fardin Ahmadizar, "A new ant colony algorithm for makespan minimization in permutationflowshops,"Applied Computers & Industrial Engineering, Vol. 63, pp.355-361, 2012.
[9] W. E. Costa, M. C. Goldbarg, E. G. Goldbarg, "New VNS heuristic for total flowtimeflowshop scheduling problem,"Applied Expert Systems with Applications, Vol. 39, pp.8149-8161, 2012.
[10] Q. K. Pen, R. Ruiz, "Local search methods for the flowshop scheduling problem with flowtimeminimization,"Applied European Journal of Operational Research, Vol. 222, pp.31-43, 2012.
[11] X. Dong, P. Chen, H. K. Huang, &Maciek Nowak, "A multi-restart iterated local search algorithm for the permutation flowshopproblem minimizing total flowtime,"Applied Computers & Operations Research, Vol. 40, pp.627-632, 2013.
[12] M. F.Tasgetiren, Q. K. Pan, P. N. Suganthan, A. H. Chen, "A discrete artificial bee colony algorithm for the total flowtime minimization in permutation flow shops,"Applied Information Sciences, Vol. 181, pp.3459-3475, 2011.
[13] C. Sangkavichit,P.Chongstitvatana, "Fragment as a Small Evidence of the Building Blocks Existence," Applied Evolutionary Learning and Optimization, Vol. 3, pp.25-44, 2010.
[14] Pei-Chann Chang, Meng-Hui Chen, Manoj K. Tiwari, AsifSikandarIquebal: A block-based evolutionary algorithm for flow-shop scheduling problem. Appl. Soft Comput. 13(12): 4536-4547 (2013).
[15] Chang, Pei-Chann, and Meng-Hui Chen. "A block based estimation of distribution algorithm using bivariate model for scheduling problems." Soft Computing (2013): 1-12.
[16] R.Agrawal, T.Imielinski and A. N.Swami,1993. Mining association rules between sets of items in large databases. In Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, P. Buneman and S. Jajodia, Eds. Washington, D.C., 207-216.
[17] Q. Zhao and S. S. Bhowmick, "Sequential pattern mining: a survey," Technical Report, CAIS, Nanyang Technological University, Singapore, No.2003118, 2003.
[18] Y. M. Chen, M.C. Chen, P. C. Chang, and S. H. Chen, "Extended artificial chromosome genetic algorithm for permutation flowshop scheduling problems," Computers & Industrial Engineering, Vol. 62, pp.536-545, 2012.
[19] ShigeyoshiTsutsui, "Probabilistic Model-Building Genetic Algorithms in Permutation Representation Domain Using Edge Histogram," Lecture Notes in Computer Science Volume 2439, pp.224-233, 2002.
[20] Chen, S. H. & Chen, M. C., "Addressing the advantage of using ensemble probabilistic models in Estimation of Distribution Algorithms for scheduling problems," Int. J. Production Economics, Vol. 141, pp. 24-33, 2013.