ICIC Express Letters
Volume **10**, Number **2**, February **2016**

ICIC International ©2016 ISSN 1881-803X
pp. **317–322**

DYNAMICAL MARKOV CHAIN COMBINED WITH TABU SEARCH AND ITS APPLICATION TO JOB SHOP SCHEDULING PROBLEM

Jianxun Li and Tomo Munehisa

Interdisciplinary Graduate School of Medicine and Engineering
University of Yamanashi
4-4-37, Takeda, Kofu, Yamanashi 400-8510, Japan
rikenkou@hotmail.co.jp; munehisa@yamanashi.ac.jp

Received July 2015; accepted September 2015

Abstract. *In order to improve the performance of the Genetic Algorithm, we have proposed a new method using the Dynamical Markov Chain Monte Carlo in previous studies. This method has been improved by extending the method to implement Simulated Annealing to the Genetic Algorithm. It is well known that the Tabu Search is another meta-heuristic approach and it can be performed for the global search problem such as a combinatorial optimization problem by using the memory structure. Therefore, in order to make the advantages of two methods work together, in this paper we will combine the Dynamical Markov Chain Monte Carlo with Tabu Search. In the experimental verification we apply this combinatorial method for the Job Shop Scheduling problem. By experimental results we confirm that it is quite effective for the optimization.*

Keywords: Genetic Algorithm, Crossover, Dynamical Markov Chain Monte Carlo, Tabu Search, Support state, Job Shop Scheduling

1. Introduction.

In general, Genetic Algorithm (GA), Simulated Annealing (SA) and Tabu Search (TS) are well used as meta-heuristics method to solve the optimization problem. GA [1] has been applied in various areas in the current society. The crossover of GA is certainly powerful mechanism, but it has lost the diversity of the solutions. By the mutation only, there is a limit by single exploration, even to maintain diversity. In SA method, introduction of temperature is effective in maintaining the diversity by Boltzmann distribution. Therefore, there has been studied about the combined methods of GA and SA such as the parallel recombined simulated annealing [2-7]. In order to make further development on these combined methods, we proposed the combining method called Dynamical Markov Chain Monte Carlo method (DMCMC) [8-11] to solve the computationally hard optimization problems. In the experiment of convergence [4] to the invariant distribution with the deceptive problem, and in the experiment of the application to the job shop schedule problem, we have proved that the DMCMC is more effective than the conventional method in a previous paper [10].

TS is another meta-heuristic method. The introduction of the tabu aims at reducing the search time without searching again the bad value that has been found. Now in order to improve the searching ability of DMCMC we make a hybrid algorithm combined with TS [12]. Therefore, how to combine them is an important issue that we need to solve. After combining the two methods, we also did the experiments to prove this hybrid algorithm is effective. And the results demonstrate that this combined method is indeed working, and it is even faster than the DMCMC method.

Contents of our paper are as follows. After a brief description on Dynamical Markov Chain Monte Carlo method and Tabu Search in Section 2, in Section 3, we also describe the method of TabuDMC, and explain how to combine TS and DMCMC. In this research we apply Job Shop Scheduling problem (JSS) [1,16] to prove our methods are effective.

Therefore, in Section 4 we illustrate the JSS, and explain how to set the Tabu List. After we make extensively experiments in Section 5, the final section is devoted to a conclusion and a discussion.

2. Materials and Methods.

2.1. **DMCMC (Dynamical Markov Chain Monte Carlo method).** DMCMC has been described extensively in a previous paper [10]. Therefore, we just describe the advantages of DMCMC here. In DMCMC by using the crossover operator we placed the different temperatures in the two populations. Therefore, (a) the support population (which has high temperature) makes the search of object population (which has low temperature) to escape from the quasi-optimal solution; (b) not only that, the support population also gives the search large amount of choices in a wide range; (c) meanwhile, by the low temperature, the object population is searching the quasi-optimal solution in detail at every feasible space until finding the optimal solution. Therefore, with those advantages the DMCMC is faster than other methods.

2.2. **TS (Tabu Search).** TS [12-14] is a technique that can perform the global search problem such as a combinatorial optimization problem. The advantages are like: (a) it can prevent the exploration terminated at quasi-optimal solutions; (b) it permits the transition to the candidate solution that is even worse than current one; (c) it can prevent the exploration from searching the same area repeatedly. These benefits are formed by equipment called Tabu List (TL) based on a history of all transitions.

Theoretically, S is the set of feasible solutions for a problem Q. For a solution $s \in S$, we calculate the cost value of s by cost function $fc : s \to C$. In TS a definition of neighborhood is very important as well as that of Simulated Annealing. Especially TS requires a lot of neighborhood and its ability depends on the method for setting neighborhood. There is also a possibility that the exploration is trapped in the same neighborhoods. Therefore, we use a function F to define a neighborhood structure and associate a set of solutions $F(s)$. From the set of solutions $F(s)$ we find the direction that starts from the solution s. Then we memorize the movements in the TL.

Neighborhood. We set N as the number of the neighborhoods of s. If the N is a large number, the improvement of solution is very fast. On the other hand, it is easy to fall into the quasi-optimal solution. Conversely, if we decrease the N, the solution can escape from the quasi-optimal immediately. However, the accuracy of the solution might be reduced greatly. Then if all neighborhoods of this state are written in the TL, the search will stop at this state. In this case the transition can get to another state all the time.

TL (Tabu List). The memory structure is crucial for a TS algorithm. In this algorithm the memory structure is expressed as the TL. We assumed that we make a transition from the current solution s to a candidate solution $s' \in F(s)$, and the bad movement in $s \to s'$ and $s' \to s$ will be written in the TL. In our recording method we write the differences between s' and s instead of writing the content of s simply. Moreover, at each transition we will record the bad movement in the TL and delete the oldest one. That is handled by "*First In First Out*" (FIFO) strategy.

3. TabuDMC.

In this section we will explain how to combine the TS with DMCMC. We named the combined one as TabuDMC. Because the advantages of two methods are complementary, purposes of this study are to prove the advantages can be combined.

In DMCMC we introduced two populations as an object state and a support state. In addition, two kinds of temperature that is characterized by SA are introduced to each population and we set the temperature separately in each population. So in the following the symbols will be introduced. Go represents a current object state; Go' represents one state of neighborhood states of the current object one; Gs represents a current support state; Gs' represents one state of neighborhood states of the current support one; To represents the temperature of object state; Ts represents the temperature of support state; the symbols of V are the cost values of corresponding states. And, $\varepsilon = \frac{Vo' - Vo}{To}$, $\varepsilon' = \frac{Vo' - Vo}{To} + \frac{Vs' - Vs}{Ts}$.

(1) Initialize the states Go, Go', Gs and Gs', and immediately calculate Vo, Vs that are the evaluation values of Go and Gs. And set the temperatures To and Ts;

(2) Memorize the contents of Go' and check in the TL
Already written: Restart step (2);
Not written: Calculate the Vo', then next step;

(3) As the evaluation (when the excellent state is that the evaluation value is the smaller one):
- *a) If $\varepsilon \leq 0$, then the contents of Go and Vo will be replaced by those of Go' and Vo';*
  - *i. If $\varepsilon < 0$, then we write the transition from Go' to Go in the TL;*
- *b) If $\varepsilon > 0$, then will generate a real number R $(0 < R < 1)$ randomly;*
  - *i. If $R > e^{-\varepsilon}$, then the contents of Go and Vo will be rewritten by those of Go' and Vo';*
  - *ii. Else, the transition from Go to Go' is written in the TL; at the same time each value remains unchanged and go to the next step;*

(4) By crossover Go and Gs, Go' and Gs' will be covered, and calculate their evaluation values Vo' and Vs';

(5) As the evaluation:
- *a) If $\varepsilon' \leq 0$, then the contents of Go, Vo and Gs, Vs will be replaced by those of Go', Vo' and Gs', Vs';*
- *b) If $\varepsilon' > 0$, then we will generate a real number R $(0 < R < 1)$ randomly;*
  - *i. If $R > e^{-\varepsilon'}$, then the contents of Go, Vo and Gs, Vs will be replaced by those of Go', Vo' and Gs', Vs';*
  - *ii. Else, each value remains unchanged and go to the next step;*

(6) If one of stopping conditions is satisfied, then the loop will be terminated. If they are not satisfied, the loop returns to step (2).

4. **Application to Job Shop Scheduling (JSS) Problem.** JSS [16-18] is a problem that determines the order of the operators that one machine processes, and minimizes the time (Makespan), when all operators finished. Below we use the work order method [16] of the JSS to carry out experiments, and the specific application method is written in the paper [10].

In the above part we have introduced the exploration principle of TS and the evaluation way of DMCMC. Therefore, the next problem is how to set the TL to record the neighborhood states of the work order method.

Tabu List (TL) for JSS. Since the capacity of the TL is limited, that is set to 5000 records after some experiments. In each record we divide it into two parts: the current state part and the neighborhood states part. We use the chromosome of the work order method to represent the current part [12]. The other one is represented by a set of transiting routes that connect the current state and the neighborhood state. Specifically, we implemented the TS only in the mutation step. Therefore, in the part of neighborhood states, we write in all the routes that we can find. And each route is written by two numbers that they are two coordinate points.

For example: *there is a 4 digits current state "1234". Because we only record the mutation step and the mutation is swapping two points, the neighborhoods of this state are expressed as (i, j), and the (i, j) express the two points that have been swapped.*

e.g., (1, 2): 2134, (1, 3): 3214, (1, 4): 4231, (2, 3): 1324, (2, 4): 1432, (3, 4): 1243. In neighborhood states part of the TL we just use the numbers in parentheses.

We update the TL by FIFO strategy. And we carried out the strategy by the searched times "N" of each current state and a search order list (SOL) of the record. In this SOL we sequentially credited the line number of each record. According to the position of each number we determine the order of records. It should be noted that the SOL only exists in one trail experiment. After the end of an exploration of the one trial, the SOL will be discarded, although the TL is used in the next trial experiment. Because the TL has the knowledge which is useful for the search and the SOL has the strong dependence on the past transitions. Then from the start of the new trial exploration, until 5000 numbers were fully recorded, we need to determine the order of update by the size of N. That is because we need to retain the large N in early exploration.

5. **Experiments.** A purpose of experiments in this section is performance comparison of DMCMC and TabuDMC methods with conventional methods. There are mainly five methods used for comparison here. Methods GA, TS and "Hybrid Genetic Algorithm and Tabu Search" (HGATS) were described at reference [14]. Especially the HGATS method has been confirmed to have an excellent searching ability. Here we use their results to compare with our methods directly. DMCMC uses the work order method (called DMCMC from here). And TabuDMC uses the work order method (called TabuDMC from here). These experiments use data that have been put in JSS benchmark [15].

5.1. **Configuration of parameters.** In order to evaluate the performance of DMCMC and TabuDMC, we should determine a number of population and two temperatures. And we have illustrated how to determine the parameters in a previous paper [11]. Therefore, we set the number of population at 8. And the two temperatures were chosen on a combination of $T_1 = 3$, $T_2 = 10$, because this choice has given us good result that is described at paper [10,11].

5.2. **Results of experiments.** Experiments are performed using kinds of JSS testing instances from the JSS benchmark [15,19]. Below we used the best values of Makespan for comparison from the experiments. The reason is that the best value can reflect the potential of each method. Depending on that the optimal is known or not, we can divide the results of instances into two categories. One is the general size category that the optimal is known as usual. In this category there are 22 kinds of instances, and these instances have been verified in previous papers [10,11]. In those papers we calculated the average values of Makespan showing the performance of DMCMC is excellent. In addition, in this paper we use the best values of Makespan, so the results of each method are very close, almost all methods have found the optimal values, so that it is difficult to determine the differences between each method. Therefore, we do not show the results of the 22 kinds of instances here. The other category is the enormous size one that the optimal is not known. And the results have shown in Table 1.

We make the table be divided into 4 parts. Column 1 specifies the instances. Column 2 specifies the size of each instance, which is expressed as the number of jobs and machines. Column 3 is composed of the UB and LB. UB is upper bound value, and LB is lower bound. Therefore, the optimal value is proved to be a value betwee n the LB and UB. And column 4 is obtained by subtracting the UB from Makespan. That is to make it easier to compare with the various methods. And column 4 is composed by the results of 5 methods. Therefore, 0 is the minimum result in column 4. A value 0 means that it has found the UB. Certainly the method is more advantageous when the result value is smaller than 0. The results of Table 1 are explained from here.

1) The results by all methods are "0" or close to "0", when they solve the instances (YN01, YN02). In these instances we can know the GA is even better than TS.

2) In instance YN03 the results did not arrive at "0" by methods GA and TS. However, the results by HGATS, DMCMC, TabuDMC, are got to "0" or close to "0". These show the excellence of HGATS, DMCMC, TabuDMC.

3) In instances SWV16, SWV17, SWV18, SWV19, SWV20, although the results by GA, TS, HGATS did not arrive at "0", the results by DMCMC, TabuDMC are got to "0". Searches by the first 3 methods are difficult to find the UB, even upper UB more than hundreds. However, by our 2 methods we have found the UB completely. These results have shown the obvious superiority of DMCMC and TabuDMC.

4) In instances SWV11, SWV12, SWV13, SWV14, SWV15, YN04, the results by all methods did not arrive at "0". Therefore, it is possible to compare the performance of each method. And the superiority of TabuDMC is represented. Especially the performances of SWV13, SWV14 and SWV15 have improved significantly.

Table 1. Results for instances of enormous size

Instances	Instance size		Optimal		Differences between UB and Makespan				
	Jobs	Machines	UB	LB	GA	TS	HGATS	DMCMC	TabuDMC
SWV11	50	10	2991	2983	209	513	21	24	14
SWV12	50	10	3003	2972	247	439	117	108	96
SWV13	50	10	3104		650	772	146	126	110
SWV14	50	10	2968		519	1038	244	127	121
SWV15	50	10	2904	2885	1331	1453	685	518	505
SWV16	50	10	2924		623	1062	402	0	0
SWV17	50	10	2794		475	665	211	0	0
SWV18	50	10	2852		28304	443	98	0	0
SWV19	50	10	2843		326	450	91	0	0
SWV20	50	10	2823		408	506	155	0	0
YN01	20	20	888	826	2	7	0	0	0
YN02	20	20	909	861	1	16	0	0	0
YN03	20	20	893	827	31	163	0	1	0
YN04	20	20	968	918	130	144	19	30	13
Average					2375.43	547.93	156.36	66.71	61.36

In the last line of Table 1, the average value of GA is extremely large, 2375.43. It means that there are some instances whose result values are still far from the optimal. And the average value of TS is 547.93, and that of HGATS is 156.36. The average values become smaller gradually. However, even for HGATS the result is unsatisfactory. Then we look at DMCMC and TabuDMC, and the results are 66.71 and 61.36, half of HGATS's result. Especially the TabuDMC's result is also the best one in five methods. Therefore, we can understand our methods have the superior performance by comparing the 5 methods.

6. **Summary.** In previous papers [10,11] we have presented the method DMCMC that is divided into two populations with different temperatures. And we have proved it to be effective. In this paper we have applied a DMCMC method to combining with TS and exert the advantages of both methods. We solved the optimization problem of JSS by this combining method. In JSS there are many operations under various constraints. With the change of each operation position, the final schedule also changes. We would like to find the optimal schedule among them. The instances of the JSS benchmark are used for experiments with five methods of GA, TS, HGATS, DMCMC and TabuDMC. The results have shown that the performance of TabuDMC is the best one among them, and the DMCMC method also has a good performance to solve the JSS problems.

In conclusion after the experimental verification we can say that the DMCMC method can be combined with TS and applied to JSS. In addition, the six advantages can play out at the same time. Throughout the whole results, the two methods (DMCMC and TabuDMC) of our research are more effective. In particular, the average value of TabuDMC is 61.36. It is smaller than 66.71 of DMCMC. Even the result values of every instance obtained by TabuDMC are smaller than those obtained by DMCMC. Therefore, in all of these methods, the performance of TabuDMC is particularly prominent. We can conclude that after lots of experiments and adjusting each parameter, the TabuDMC method will obtain some better results. In the future in order to broaden the scope of TabuDMC, we will continue to study the TabuDMC with other optimization problems.

REFERENCES

[1] M. D. Vose, *The Simple Genetic Algorithm*, The MIT Press, London, 1999.

[2] S. W. Mahfoud and D. E. Goldberg, Parallel recombinative simulated annealing: A genetic algorithm, *Parallel Computing*, vol.21, no.1, pp.1-28, 1995.

[3] S. W. Mahfoud, Parallel recombinative simulated annealing: A genetic algorithm, *IlliGAL Report No. 93006*, 1993.

[4] R. Azencott, *Simulated Annealing Parallelization Techniques*, John Wiley & Sons, New York, 1992.

[5] S. W. Mahfoud and D. E. Goldberg, A genetic algorithm for parallel simulated annealing, *Parallel Problems Solving from Nature*, vol.2, pp.301-310, 1992.

[6] E. H. L. Aarts and J. H. M. Korst, *Simulated Annealing and Boltzmann Machines*, John Wiley & Sons, New York, 1989.

[7] P. D. Morail and L. Miclo, On the convergence and applications of generalized simulated annealings, *SIAM J. Control Optim.*, vol.37, pp.1222-1250, 1999.

[8] H. Morimura and Y. Takahashi, *Markovian Analysis*, Nikkagiren, Tokyo, 2000.

[9] T. Munehisa, K. Yamamoto and J. Li, Markov chain Monte Carlo with dynamical state generation, *SIG-FPAI*, pp.87-92, 2011.

[10] J. Li and T. Munehisa, Genetic algorithm using the inhomogeneous Markov chain for job shop scheduling problem, *ICIC Express Letters*, vol.9, no.2, pp.501-509, 2015.

[11] J. Li and T. Munehisa, Markov chain Monte Carlo method with dynamical state generation and its application to JSS, *JPNSEC*, vol.1, no.1, 2010.

[12] M. Dell'Amico and M. Trubian, Applying tabu search to the job-shop scheduling problem, *Annals of Operations Research*, vol.41, pp.231-252, 1993.

[13] R. Thamilselvan and P. Balasubramanie, Integrating genetic algorithm, tabu search approach for job shop scheduling, *International Journal of Computer Science and Information Security*, vol.2, no.1, 2009.

[14] R. Thamilselvan and P. Balasubramanie, Integration of genetic algorithm with tabu search for job shop scheduling with unordered subsequence exchange crossover, *Journal of Computer Science*, vol.8, no.6, pp.681-693, 2012.

[15] *CP2SAT: JSSbenchmark Results*, http://bach.istc.kobe-u.ac.jp/csp2sat/jss/, 2010.

[16] R. Cheng, M. Gen and Y. Tsujimura, A tutorial survey of job shop scheduling problems using genetic algorithms-I: Representation, *Computers & Industrial Engineering*, vol.30, no.4, pp.983-997, 1996.

[17] R. Cheng, M. Gen and Y. Tsujimura, A tutorial survey of job shop scheduling problems using genetic algorithms-II: Hybrid genetic search strategies, *Computers & Industrial Engineering*, vol.36, pp.343-364, 1999.

[18] R. Zhang and C. Wu, A hybrid approach to large-scale job shop scheduling, *Applied Intel.*, vol.32, pp.47-59, 2008.

[19] T. Yamada and R. Nakano, Scheduling by genetic local search with multi-step crossover, *Proc. of the 4th International Conference on Parallel Problem Solving from Nature*, Berlin, Germany, pp.960-969, 1996.