

Tabu Exponential Monte-Carlo with Counter Heuristic for Examination Timetabling

Nasser R. Sabar¹, Masri Ayob² and Graham Kendall³

^{1,2}*Jabatan Sains Komputer, Fakulti Teknologi dan Sains Maklumat,
Universiti Kebangsaan Malaysia, 43600 UKM, Bangi Selangor.
naserdolayme@yahoo.com, masri@ftsm.ukm.my*

³*ASAP Research Group, School of Computer Science, The University of Nottingham, Nottingham NG8 1BB, UK.
gxx@cs.nott.ac.uk*

Abstract—In this work, we introduce a new heuristic TEMCQ (Tabu Exponential Monte-Carlo with Counter) for solving exam timetabling problems. This work, an extension of the EMCQ (Exponential Monte-Carlo with Counter) heuristic that was originally introduced by Ayob and Kendall. EMCQ accepts an improved solution but intelligently accepts worse solutions depending on the solution quality, search time and the number of consecutive non-improving iterations. In order to enhance the EMCQ heuristic, we hybridise it with tabu search, in which the accepted moves are kept in a tabu list for a certain number of iterations in order to avoid cyclic moves. In this work, we test TEMCQ on the un-capacitated Carter’s benchmark examination timetable dataset and evaluate the heuristic performance using standard proximity cost. We compare our results against other methodologies that have been reported in the literature over recent years. Results demonstrate that TEMCQ produces good results and outperforms other approaches on several benchmark instances.

I. INTRODUCTION

Examination timetabling deals with the allocation of a number of exams to a given number of timeslots while satisfying a set of constraints (see Carter and Laporte [1]). The constraints for exam timetabling are typically divided into two categories: hard and soft. Hard constraints are rigidly enforced and cannot be violated under any circumstances. For example, there cannot be any students sitting two exams at the same time. A timetable that is able to satisfy all of the hard constraints is feasible. Soft constraints, such as allowing sufficient time between exams in order to give students enough time to revise, should be satisfied as far as possible. Soft constraints may vary from one academic institution to another [2]. However, due to the complexity of the problem, it is not usually possible to generate solutions without violating some of these soft constraints. In order to minimise a violation of soft constraints, a penalty cost is attached to each violation. The measurement of the quality of an exam timetable is based on the value of the penalty cost given. A lower penalty shows a better quality of exam timetable.

To date, many approaches have been designed to solve

timetabling problem. For example: graph based heuristics [3][4], basic local search [5][6][7], tabu search [8][9], simulated annealing ([10]), genetic algorithms [11], memetic algorithms [2][12] great deluge algorithms [13], Ant Colony[14], particle swarm optimization[15] and Fuzzy Reasoning[16]. These methodologies have been successfully applied to timetabling problems. Further information about exam timetabling can be found in [17-22].

In this work, we hybridise tabu search and EMCQ heuristics in order to take advantage of *tabu list* that can memorise recent moves. When a new search is carried out, it will search for a solution space by excluding neighbours that have restricted by a *tabu list*. When designing tabu search, it is important to fine-tune the parameters according to the problem in hand. Such parameters include the *tabu tenure* and the stopping conditions.

In 2007, Abdullah et al. [23] developed a large neighbourhood search for different optimization problems which was based on the improvement graph construction methodology, originally developed by Ahuja and Orlin [30]. A tree-based neighbourhood structure was designed to carry out cyclic exchanges among the timeslots, rather than the more traditional pair-wise exchange based operators that were more commonly employed. Then, they stored improved moves in a *tabu list* for a certain number of *Tabu tenure*. This approach reported the best results on several Carter’s dataset problems at the time of its publication. However, one of the limiting factors of applying this method in solving exam timetabling problems is the fact that it requires a great deal of computational time to generate a solution.

In 2001 White and Xie [31] developed a four-stage Tabu Search called OTTABU. Using an exam timetabling problem from the University of Ottawa, their approach showed that solutions could be gradually improved by considering a greater number of constraints at each stage of the process. In addition to utilising recent short term memory, long term memory was also used to record the frequency of the most active moves in the search history. They determined the size of the long term memory by analyzing the less important exams in the problem first.

Based on the capability of memorising previous moves

using the *tabu list*, in this work, we propose a new heuristic (TEMCQ) that embeds a tabu list within the EMCQ (Exponential Monte-Carlo with Counter [29]) algorithm. We utilise graph coloring heuristics (hybridisations of *Least Saturation Degree*, *Largest Degree first* and *Largest Enrolments First* heuristics) in constructing the initial solution (see Ayob et al. [28]). The solution is then improved by an adaptive TEMCQ heuristic search. The aim of this work is to investigate the effectiveness of combining a tabu list with EMCQ in searching for good quality solutions. In order to demonstrate the performance of TEMCQ heuristic, we test the heuristic on un-capacitated (where the size of the room is disregarded) Carter’s [27] examination timetabling datasets (type I, see [18]). In this work, we use standard proximity cost [27] to evaluate the quality of timetable.

II. TABU EMCQ HEURISTIC

EMCQ (Exponential Monte-Carlo with Counter) heuristic was introduced by Ayob and Kendall [29]. The EMCQ algorithm is quite similar to a simulated annealing approach. The difference is that, no cooling schedule is required. EMCQ always accepts improving solutions. A non-improving solution is adaptively accepted based on the solution quality, search time and the number of consecutive non-improving iterations (i.e. a period where the search is trapped in a local optima). In order to accept the non-improving moves, a random number $[0, 1]$ is generated. The move is accepted if the generated random number is less than $e^{-\Theta/\lambda}$ where $\Theta = \delta * t$, $\lambda = P(Q)$. δ is the difference between the previous and trial solutions (i.e. $\delta = f(Sol^*) - f(Sol)$ where $f(Sol^*)$ and $f(Sol)$ are the quality of the trial solutions and previous solution, respectively) and t is an iteration counter. $P(Q)$ is a function to intelligently control Q , where Q is the number of consecutive non improving moves. This work uses $\lambda = Q$ (selected based on preliminary experiments) in order to adaptively control the acceptance probability. The probability of accepting a worse solutions decreases as the number of iterations increase. However, if there is no improvement over a series of consecutive iterations, then the probability of accepting worse solution will increase accordingly with regard to the quality of the trial solution, as well as the search time.

In this work, we enhance the EMCQ heuristic by hybridising it with a tabu search algorithm. We utilise a short term memory in tabu search in order to improve the search process in the EMCQ heuristic. Any event (i.e. exams in this case) that have been involved in a move are added to the tabu list (in this work, it has a length of 4, 6 and 8). The exams in the tabu list are prohibited to be chosen for the next move for a certain number of iterations. Thus, we give more opportunity to other exams to be considered in performing moves that may result in better quality solutions and attempt to avoid cyclic moves. All examinations in the *tabu list* will change to non tabu status after a limited number of successful iterations are reached (i.e. accepted moves). We

set the *tabu tenure value based on preliminary experiments*.

Let us consider a solution space S ; an objective function f and a neighborhood structure n . Figure I present the pseudo-code for our proposed method, TEMCQ.

```

Generate Initial Solution, Sol; calculate the initial cost, f (Sol);
Set best solution, Solbest ← Sol; f (Solbest) ← f (Sol);
Set maximum number of iterations, Max-Iter=MAX_ITER; Maximum
tabu length, MaxT_Length=MAX_T_LENGTH; Maximum
consecutive Un-improvement, Max_Un-improvement =MAX_Un-
improvement;
Set Tabu list length, T_Length=0; Tabu tenure, T_Tenure=0; Q=1;
Number of Iteration, T=0; Un-improvement =0;
Do while (T < Max-Iter)
{
  Generate a feasible neighbor solution, Sol* that is not restricted by
  a tabu list and calculate f (Sol*);
  Calculate  $\delta = f(Sol^*) - f(Sol)$ ;
  If (f (Sol*) <= f (Sol))
  {
    Sol ← Sol*; f (Sol) ← f (Sol*);
    Q=1; Un-improvement=0;
    Remove first item from the tabu list if T_Length >=
    MaxT_Length
    Add new move into the end of tabu list; T_Length++;
    If (f (Sol*) <= f (Solbest))
      { Solbest ← Sol; f (Solbest) ← f (Sol) ;}
  }
  Else
  {
    Generate a random number R between [0, 1];
    If (R ≤ e- $\delta * T / Q$ )
    {
      Sol ← Sol*; f (Sol) ← f (Sol*);
      Q=1; Un-improvement=0;
      Pop out the first item from the tabu list if T_Length >=
      MaxT_Length
      Add new move into the end of tabu list; T_Length++;
    }
    Else
    {
      Un-improvement= Un-improvement+1;
      If (Un-improvement== Max_Un-improvement)
      {
        Q=Q+1; Un-improvement=0 ;}
      }
  }
  T=T+1;
} end while;
Return the best solution, Solbest and f (Solbest ).

```

Figure I: TEMCQ algorithm

The algorithm starts by generating a feasible, initial solution. In this work, we construct an initial solution using graph coloring heuristics that hybridise *least saturation degree*, *largest degree First* and *largest enrolments First* (LS+LD+LE) heuristics proposed in Ayob et al. [28]. Then, we initialise all the parameters. Next, we iteratively generate a feasible trial solution. The feasible trial solution is a new solution, that is not restricted by *Tabu List* and that satisfies all the hard constraints. In this work, a trial solution is generated by randomly moving an event (exam) to another clash free timeslot. Once a new feasible solution, Sol^* is obtained, we calculate the quality of the new solution and compare with the quality of the incumbent solution. The quality of the solution is measured using the standard Proximity Cost (see [27]). If there is an improvement (or even make the same quality), we accept the new solution and replace it with the incumbent solution (i.e. $Sol \leftarrow Sol^*$), then we also update the best solution ($Sol_{best} \leftarrow Sol$; $f(Sol_{best}) \leftarrow f(Sol)$) if $f(Sol^*) \leq f(Sol_{best})$. We also accept the same quality solution because the new solution is different from the old solution and accepting it might help avoid becoming trapped in a local optima.

However, in order to escape from a local optima, a worse solution is adaptively accepted with a probability that depends on Q (the number of consecutive non-improvement iterations), solution quality (δ) and search time (t). A worse solution is more likely to be accepted if δ is small or Q is large. This is a diversification factor where the search will diversify when it is trapped in a local optima. The intensification factor is controlled by a parameter T . That is, the larger the value of T , the more unlikely we are to accept a worse solution.

In this work, we increase Q by 1 when a limited number of consecutive, non-improving moves (the number of consecutive non-improving was chosen based on preliminary experiments) are reached. The idea of TEMCQ is to ensure that we only accept a moderately worse solution after most of the neighborhood of the current solution has been explored and none of them are found to be better than the old solution. The accepted move will be added to the end of *Tabu List* and we remove the first item from the tabu list if its length is greater than a pre-defined number. That is, each exam will stay in tabu list for a certain number of accepted moves (tabu tenure) in order to avoid cyclic moves. This procedure is repeated until we reach the number of iterations (Max-Iter).

III. EXPERIMENTS AND RESULTS

The algorithm was tested on Carter’s un-capacitated exam timetabling benchmark datasets [27] (type I, [18]). We perform our experiment on 12 datasets (10 runs for each datasets). The characteristics of datasets are shown in Table I. Our algorithm was implemented using Visual C++ and tested on a PC with an AMD Athlon 1.92 GHz processor, 512 RAM and Windows XP 2002.

The results of this experiment (best result out of ten runs) are presented in Table 2 and are comparable to the

current state-of-the-art approaches reported in the literature. The best results obtained from literature and our works are shown in bold.

In this work, *tabu tenure* and the limited number of consecutive non-improving were selected after thorough

TABLE I
UN-CAPACITATED STANDARD CARTER BENCHMARK EXAM TIMETABLING DATASET

DATASETS	Number of timeslots	Number of examinations	Number of students
Car-f-92	32	543	18419
Car-s-91	35	682	16925
Ear-f-83	24	190	1125
Hec-s-92	18	81	2823
Kfu-s-93	20	461	5349
Lse-f-91	18	381	2726
Rye-s-93	23	486	11483
Sta-f-83	13	139	611
Tre-s-92	23	261	4360
Uta-s-92	35	622	21267
Ute-s-92	10	184	2750
Yor-f-83	21	181	941

testing for each instance. We have tested number of different alternative values and the ones selected are those that gave the best results concerning both the quality of the solution and the computational time needed to achieve this solution. Thus, the selected *tabu tenure* values are 4, 6 and 8 whilst the limited numbers of consecutive non-improving are 10, 10, and 8 respectively. The best result over the 12 datasets is obtained when the value of *tabu tenure*=4, the limited number of consecutive non-improving=10 for (Car-f-92, Car-s-91, Tre-s-92, Uta-s-92, Yor-f-83), *tabu tenure*=6, the limited number of consecutive non-improving=10 for (Ear-f-83, Hec-s-92) and *tabu tenure*=8, the limited number of consecutive non-improving=8 for (Kfu-s-93, Rye-s-93, Ute-s-92) respectively .

TEMCQ outperforms other well known approaches that have been reported in the literature (with regards to examination timetabling) in solving five instances (i.e. Car-f-92, Car-s-91, Tre-s-92, Uta-s-92 and Yor-f-83). However, Although TEMCQ shows the worst solutions for (Rye-s-93), we belief, TEMCQ might be able to produce good quality solutions (comparable to other well-known approaches) if parameter modifications are applied such as changing *tabu tenure*, *tabu list* length or the number of consecutive non-improvements. This will be reported in our future work after performing further experiments.

TABLE II
RESULTS OBTAINED FROM TABU EMCQ HEURISTICS COMPARED TO THE LITERATURE

Dataset	Carter et al. [27]	Di Gaspero & Schaerf [8]	Caramia et al. [6]	Burke & Newall [5]	Merlot et al. [7]	Kendall & Hussin [9]	Asmuni et al. [16]	White et al. [31]	Burke et al. [17]	Burke et al. [4]	Burke et al. [21]	Abdullah et al. [26]	Abdullah et al. [20]	TEMQC
Car-f-92	6.2	5.2	6.0	4.10	4.3	4.67	4.56	4.63	4.4	5.36	4.6	4.4	4.1	4.09
Car-s-91	7.1	6.2	6.6	4.65	5.1	5.37	5.29	5.73	4.8	4.53	4.0	5.2	4.8	4.00
Ear-f-83	36.4	45.7	29.3	37.05	35.1	40.18	37.02	45.8	35.4	37.92	32.8	34.9	36.0	41.21
Hec-s-92	10.8	12.4	9.2	11.54	10.6	11.86	11.78	12.9	10.8	12.25	10.0	10.3	10.8	12.34
Kfu-s-93	14.0	18.0	13.8	13.90	13.5	15.84	15.81	17.1	13.7	15.2	13.0	13.5	15.2	16.46
Lse-f-91	10.0	15.5	9.6	10.82	10.5	-	12.09	14.7	10.4	11.33	10.0	10.2	11.9	14.50
Rye-s-93	7.3	-	6.8	-	8.4	-	10.35	11.6	8.9	-	-	8.7	-	12.00
Sta-f-83	161.5	160.8	158.2	168.73	157.3	157.38	160.42	158	159.1	158.19	159.9	159.2	159.0	159.20
Tre-s-92	9.6	10.0	9.4	8.35	8.4	8.39	8.67	8.94	8.3	8.92	7.9	8.4	8.5	7.89
Uta-s-92	3.5	4.2	3.5	3.20	3.5	-	3.57	4.44	3.4	3.88	3.2	3.6	3.6	3.19
Ute-s-92	25.8	29.0	24.4	25.83	25.1	27.60	27.78	29.0	25.7	28.01	24.8	26.0	26.0	28.76
Yor-f-83	41.7	41.0	36.2	37.28	37.4	-	40.66	42.3	36.7	41.37	37.28	36.2	36.2	36.19

IV. CONCLUSIONS

We have presented a new heuristic, Tabu EMCQ (TEMQC) that hybridizes tabu search with Exponential Monte-Carlo with Counter (EMCQ). TEMQC always accepts improving solutions. Worse solutions are adaptively accepted based on the quality of a trial solution, the search time and the number of consecutive non-improving iterations. That is, a trial solution may be accepted based on the EMCQ's acceptance criterion. If this happens, a new solution must be sought by exploring new neighborhoods. The EMCQ method will check the probability of accepting a worse solution. This may prevent the search from becoming trapped for a long time depending on the number consecutive non-improving moves and the quality of the trial solution. In this case, the parameters can be intelligently controlled by automatically increasing the desired number of the non-improvement counter, Q.

The accepted moves will be added to Tabu List to avoid cycling moves during the search process. In addition, this restriction could give more opportunity to other events to be considered in performing moves that may result in better quality solutions. Results demonstrate that TEMQC is very promising and can be used to produce good quality solutions that are comparable to other published results (for solving examination timetabling). This work also demonstrates that embedding a tabu list with EMCQ has produced a good heuristic search. We believe that TEMQC could also be applied to other problem domains.

REFERENCES

- [1] M.W. Carter and, G., Laporte (1996). Recent developments in practical examination timetabling. In: Burke and Ross (1996), 3-21.
- [2] E.K. Burke, J.P. Newall and R.F. Weare. (1996b). A memetic algorithm for university exam timetabling. The Practice and Theory of Automated Timetabling I: Selected Papers from first International Conference on the Practice and Theory of Automated Timetabling (PATAT I), Edinburgh, UK, Lecture Notes in Computer Science 1153, Springer-Verlag. (Editors: E.K. Burke and P. Ross), pp 241-250.
- [3] D. de Werra. (1985). an introduction to timetabling - European Journal of Operational Research, 19, 151-162.
- [4] E.K. Burke, B. McCollum, A. Meisels, S. Petrovic and R. Qu (2007). A graph based hyper-heuristic for exam timetabling problems. European Journal of Operational Research, 176:177-192, 2007.
- [5] E.K., Burke, J.P. Newall Enhancing Timetable Solutions with Local Search Methods. In: Burke, E, Causmaecker, P.D. (Eds.): Practice and Theory of Automated Timetabling IV (PATAT 2002, Gent Belgium, August, selected papers). Lecture Notes in Comp. Science, Vol. 2740. Springer-Verlag, Berlin Heidelberg New York (2003) 195-206.
- [6] M., Caramia, P., Dell'Olmo, G. F. Italiano, New algorithms for examination timetabling. In: Naher, S., Wagner, D. (Eds.): *Algorithm Engineering 4th Int. Workshop, Proc. WAE 2000 (Saarbrücken, Germany, September) Lecture Notes in Computer Science*, Vol. 1982. Springer-Verlag, Berlin Heidelberg New York (2001) 230-241.
- [7] L.T.G Merlot, N. Boland, B.D., Hughes, P.J Stuckey, A hybrid algorithm for examination timetabling problem. In: Burke, E, Causmaecker, P.D. (Eds.): *Practice and Theory of Automated Timetabling IV (PATAT 2002, Gent Belgium, August, selected papers)*. Lecture Notes in Computer Science, 2740. Springer-Verlag, Berlin Heidelberg New York (2003) 207-231.
- [8] L., Di Gaspero, Schaerf, A. Tabu search techniques for examination timetabling. In: Burke, E., Erben, W. (Eds.): Practice and Theory of Automated Timetabling III (PATAT 2000, Konstanz Germany, August, selected papers). Lecture Notes in Computer Science, Vol. 2079. Springer-Verlag, Berlin Heidelberg New York (2001) 104-117.
- [9] G. Kendall, and N. M. Hussin, Tabu Search Hyper-Heuristic Approach to the Examination Timetabling Problem at University Technology MARA. In: E. K. Burke & M. Trick (Eds): *Proceedings of the 5th International Conference on Practice and Theory of Automated Timetabling (PATAT 2004)*, Pittsburgh, USA, August 2004, pages 199-217.
- [10] J.M., Thompson, K.A Dowsland, A robust simulated annealing based examination timetabling system. *Computers and Operations Research*. 25 (1998) 637-648.
- [11] E.K., Burke, D.G. Elliman, and R.F. Weare, A hybrid genetic algorithm for highly constrained timetabling problems. *Proceedings of the 6th International Conference on Genetic Algorithms (ICGA'95, Pittsburgh, USA, 15th-19th July 1995)*. (1995) 605-610, Morgan Kaufmann, San Francisco, CA, USA.
- [12] E.K. Burke and J.D. Landa Silva. (2004). The design of memetic algorithms for scheduling and timetabling problems. Recent Advances in Memetic Algorithms, Studies in Fuzziness and Soft Computing, (Editors: N. Krasnogor, W. Hart and J.Smith), Springer, 166, pp 289-312.
- [13] E.K., Burke, Y., Bykov, J., Newall, S. Petrovic, A time-predefined local search approach to exam timetabling problems. *IIE Transactions on Operations Engineering* 36 (2004) 509-528

- [14] K.A. Dowsland and J.M. Thompson. (2005). Ant colony optimisation for the examination scheduling problem. *Journal of the Operational Research Society*, 56, pp 426-438.
- [15] S.C., Chu, Y.T Chen, and J.H. Ho, Timetable scheduling using particle swarm optimization. In first international conference on innovation computing, Information and control (2006), Vol. 3, pp. 324-327.
- [16] H. Asmuni, E.K. Burke, J.M. Garibaldi and B. McCollum. (2005a). Fuzzy multiple ordering criteria for examination timetabling. *The Practice and Theory of Automated Timetabling V: Selected Papers from 5 International Conference on the Practice and Theory of Automated Timetabling (PATAT V)*, Pittsburg, USA, Lecture Notes in Computer Science 3616, Springer-Verlag, (Editors: E.K. Burke and M. Trick), pp 334-353.
- [17] E.K. Burke, J. Kingston and D. de Werra. (2004a). Applications to timetabling. *Handbook of Graph Theory*, (Editors: J. Gross and J. Yellen), Chapman Hall/CRC Press, pp 445-474.
- [18] R. Qu, E. K. Burke, B. McCollum, L.T.G. Merlot, and S.Y. Lee. *A Survey of Search Methodologies and Automated System Development for Examination Timetabling*. To appear in *Journal of Scheduling*, 2008. doi: 10.1007/s10951-008-0077-5.
- [19] E.K. Burke and S. Petrovic. (2002). Recent research directions in automated timetabling. *European Journal of Operational Research*, 140(2), pp 266-280.
- [20] S. Abdullah and E.K. Burke. A Multi-start Large Neighbourhood Search Approach with Local Search Methods for Examination Timetabling, accepted for publication in the *Proceedings of the International Conference on Automated Planning and Scheduling (ICAPS 2006)*, Cumbria, UK, June 6th-June 10th, 2006.
- [21] E.K. Burke, A.J. Eckersley, B. McCollum, S. Petrovic and R. Qu. (2006). Hybrid variable neighbourhood approaches to university exam timetabling. *Metaheuristic International Conference*, 25-29 June 2007, Montreal.
- [22] M. Ayob, S. Abdullah and A. M. A. Malik (2007). A Practical Examination Timetabling Problem at the Universiti Kebangsaan Malaysia,. *International Journal of Computer Science and Network Security*, 7(9), pp 198-204.
- [23] S. Abdullah, S. Ahmadi, E.K. Burke, M. Dror and B.McCollum. (to appear 2006). A tabu-based large neighbourhood search approach for the capacitated examination timetabling problem. *Journal of the Operational Research Society*, Volume 58 (01 Nov 2007), pp 1494 – 1502.
- [24] M. Ayob, E.K. Burke and G. Kendall. (2006). An iterative re-start variable neighbourhood search for the examination timetabling problem. Extended Abstract. *Proceedings of the 6th International Conference on the Practice and Theory of Automated Timetabling*. 30th August-1st September 2006, Brno, The Czech Republic, pp 336-344.
- [25] E.K. Burke and Y. Bykov. (2006). Solving exam timetabling problems with the flex-deluge algorithm. Extended Abstract. *Proceedings of the 6th International Conference on the Practice and Theory of Automated Timetabling*. 30th August-1st September 2006, Brno, The Czech Republic, pp 370-372.
- [26] S. Abdullah, S. Ahmadi, E.K. Burke and M. Dror, (2007). Investigating Ahuja-Orlin's Large Neighbourhood Search Approach for Examination Timetabling. *OR Spectrum* 29 (2), 351-371.
- [27] M. W., Carter, G. Laporte, and S. Y. Lee, (1996) Examination timetabling: algorithmic strategies and applications. *Journal of the Operational Research Society* Volume 47 Issue 3, 373-383.
- [28] M. Ayob, A. M. A. Malik, S. Abdullah, A. R. Hamdan, G. Kendall, and R. Qu (2007). Solving a Practical Examination Timetabling Problem: A Case Study". In O. Gervasi and M. Gavrilova (Eds.): *ICCSA 2007, Lecture Notes In Computer Science 4707, Part III*, Springer-Verlag Berlin Heidelberg, pp. 611–624.
- [29] M. Ayob and G. Kendall. (2003). A Monte Carlo hyper-heuristic to optimize component placement sequencing for multi head placement machine. *The Proceedings of the International Conference on Intelligent Technologies (InTech'03)*, Chiang Mai, Thailand, pp 132-141.
- [30] R.K. Ahuja, J.B. Orlin and D. Sharma (2001). Multi-exchange neighbourhood search algorithm for capacitated minimum spanning tree problem. *Mathematical Programming*, 91: 71-97.
- [31] G.M. White and B.S. Xie (2001). Examination timetables and tabu search with longer-term memory. In: E.K. Burke and W. Erben (Eds) (2001). *Practice and Theory of Automated Timetabling: Selected Papers from the 3rd International Conference*. Lecture Notes in Computer Science, 2079. 85-103.