# A Comparison between Hybrid Population-based Approaches for solving Post-Enrolment Course Timetabling Problems

**Ghaith M. Jaradat and Masri Ayob**,

Data Mining and Optimization Research Group, Centre of Artificial Intelligence Technology,
The National University of Malaysia, Faculty of Information Science and Technology, 43600 UKM Bangi, Malaysia

**Summary**
In this study, we apply our Elitist-Ant System, Big Bang-Big Crunch and Scatter Search heuristics to solve two post-enrolment course timetabling problems (first and second international timetabling competitions) and to compare their performance and consistency. The approaches mainly focus on employing the elite pool and solution combination strategies. Both strategies provide deterministic search guidance by maintaining a balance between diversity and quality of the population. This is achieved by a dynamic changing of the population size and, the utilization of elite solutions and a probabilistic selection procedure in generating good quality and diversity solutions. Experimental results showed that our hybrid approaches produce good quality solutions, and outperforms some best known results reported in the literature including population-based algorithms. In term of solutions' quality, the Scatter Search ranked first and followed closely by the Elitist-Ant System and Big Bang-Big Crunch heuristics.

***Key words:***
*Elitist-Ant System, Big Bang-Big Crunch, Scatter Search, post-enrolment course timetabling problem, elite pool, solution combination.*

## 1. Introduction

The university course timetabling problem is considered as NP-hard problem [1], which is difficult to solve for optimality. During the last decade, various metaheuristics have been applied to solve course timetabling problem (see [2]). Metaheuristics are classified into two classes, population-based and single-based (aka local search) metaheuristics [3]. Some common population-based methods applied to the problem are the ant colony optimization [4] [5] [6], memetic algorithm [7] [8] and hybrid evolutionary algorithm [9]. Mainly, the population-based metaheuristics are intensively investigated, where the population-based metaheuristics are utilized due to their capability of search space exploration and can be easily combined with local search methods to enhance the solution exploitation process [10]. Whilst, some common single-based methods applied to the problem are tabu search [5], simulated annealing [5], dual sequence simulated annealing with round-robin [11] and great deluge with kempe chain neighbourhood structure [12]. The single-based metaheuristics are utilized due to their capability of solution space exploitation.

The strength of population-based methods is certainly based on the capability of recombining solutions to obtain new ones [3]. In Evolutionary algorithms (EA) including scatter search, explicit recombinations (which are move and swap of assignments in a solution representing information exchange between generations of a solution's good components) are implemented by one or more recombination operators, such as crossover and mutation [3]. Whilst, in Ant Colony Optimization (ACO), recombination is implicit, i.e. new solutions are generated using a distribution over the search space which is a function of earlier populations representing the search experience [3]. The implicit recombination enables the search process to perform a guided sampling of the search space [3]. Both recombination techniques can effectively find promising areas of the search space [3].

However, a population-based metaheuristic is considered weak in intensifying the search for higher quality solutions. Hence, in order to enhance the intensification process, a specialized metaheuristics in exploiting the solution space (e.g. hill climbing) is usually hybridized with the population-based metaheuristics. Many studies have recommended the hybridization between a population-based metaheuristic and other single-based metaheuristics, such as [10] [13] [14]. Local search metaheuristics are able to overcome the weakness (in the population-based) of exploiting the solution space (further enhancement of a solution's quality).

Jaradat and Ayob [15] enhanced the capability of the Elitist-Ant System in maintaining a balance between diversification and intensification of the search for solving the course timetabling problem. This is achieved by hybridizing the Elitist-Ant System with an iterated local search and an intensification mechanism to intensify the search further. A diversification mechanism is also employed to escape the local minima. Jaradat and Ayob

Manuscript received November 5, 2011
Manuscript revised November 20, 2011

[16] enhanced the Big Bang-Big Crunch metaheuristic to solve the course timetabling problem. The approach utilizes an elite pool of good quality solutions and a dynamic population size to produce good quality solutions. Recently, Jaradat and Ayob [17] applied a hybrid Scatter Search to the course timetabling problem. The approach mainly utilizes a collection of elite solutions (containing good quality and diverse solutions) in generating new good quality solutions.

In this work, we focus on the university post-enrolment course timetabling problems (namely the 1st and 2nd timetabling competitions). This work mainly aims at comparing the performance of three hybrid population-based approaches: hybrid Elitist-Ant System (Elitist-AS), hybrid Big Bang-Big Crunch optimization (BB-BC) and hybrid Scatter Search (SS) in solving the post-enrolment course timetabling problems. These approaches were proposed in [15] [16] [17]. In the previous works we test these approaches only on the Socha's benchmark datasets [18].

## 2. Description of the Problem

Course timetabling problems mainly comprise of assigning a set of courses, students and lecturers to a specific and fixed number of timeslots and rooms in a week, while satisfying some constraints [19]. In this work, we tested our hybrid approaches on benchmark post-enrolment course timetabling instances of the 1st timetabling competition (TTComp2003, [20]) which consider only student preferences. These instances were generated by the Metaheuristic Network (refer to the official website). The 2nd international timetabling competition (ITC2007-Track2, [21]) is also considered in our experiments [2]. The benchmark problems are formulated as follows:

- A set of *N* courses needs to be scheduled into 5 working days a week of 9 timeslots each day, where *T*=45 timeslots,
- A set of *R* rooms is given, where each room has a number of *F* features that include their capacities and other facilities,
- A number of *M* students will attend the course. Each student attends a number of courses with a given size of each room involved.

There are two types of constraints: hard and soft. In order to produce a feasible timetable, all of the hard constraints must be satisfied, whereas the violation of the soft constraints must be minimized in order to produce a good quality timetable. Each violation of soft constraints will incur a penalty cost, where lower penalty values indicate good quality solutions. A feasible timetable is one in which all courses have been assigned to timeslots and rooms, and all hard constraints are satisfied. The hard constraints for both competitions' instances are:

*H1*: No student attends more than one course at the same time;
*H2*: The room is big enough for all the attending students and satisfies all the features required by the course;
*H3*: Only one course is scheduled in each room at any timeslot;
*H4*: Events are only assigned to timeslots that are pre-defined as available for those events (applicable only to ITC2007-Track2);
*H5*: where specified, events are scheduled to occur in the correct order in the week (applicable only to ITC2007-Track2);

Then, a quality of timetable is measured by penalising equally each violation of the following soft constraint (i.e. penalty cost=1 for each violation). The soft constraints for the problem are:

*S1*: A student should not has a class in the last slot of the day;
*S2*: A student should not has more than two classes consecutively;
*S3*: A student should not has a single class on a day.

The objective function value of a timetable for each student is simply calculated as the summation of the hard and soft constraints violations (as in [5]). However, we deal only with feasible solutions in our approaches. More information about the instances and the problem formulation can be found in [2] [18].

## 3. The Hybrid Approaches

In this work, we extend the investigation of our hybrid approaches [15] [16] [17] by applying them to the TTComp2003 and ITC2007-Track2 instances.

### 3.1 Hybrid Elitist-Ant System

The Elitist-AS was originally proposed by [22]. Since, the Elitist-AS is incapable of maintaining a balance between diversity and quality of the search, in our previous work [15] we enhanced its capability by hybridizing the Elitist-AS with an Iterated Local Search (ILS), diversification and intensification mechanisms and, an external memory to store elite solutions. The ILS is employed for a significant solution improvement by accepting only a better solution. The diversification mechanism is employed by restarting the ant search (when it stagnates) to explore different regions (when no further possible improvement) of the search space. Whilst, the intensification mechanism is employed to further explore the neighborhoods of a solution further. Those diversification and intensification mechanisms help in strengthen the ability of the pheromone deposition (intensification) and evaporation (diversification) in diversifying the search while maintaining the quality. Based on a collection of diverse elite solutions stored in an external memory and the number of non-improvement iterations (in the ILS), the intensification mechanism will be activated and commenced the search in improving the solution obtained from the ILS further. Whilst, the diversification mechanism will be activated and commenced when the ILS fails to improve the quality of solutions and the pheromone trails will be reinitialized. The pseudo code of our hybrid Elitist-AS is illustrated in Fig. 1 [15]. Our hybrid Elitist-AS algorithm starts the search by constructing a population of initial solutions using a constructive heuristic.

```
Initialization phase ();
while StoppingCriterion not met do
Construction phase ();
    for each ant // solution construction
        Assign all course into feasible timeslots & rooms using
          A probabilistic rule;
    end for
Improvement phase ();
    while non-improvement stopping criterion not met do
        Locally improve each constructed solution; //employ ILS
        Update size & content of external memory;
    end while
if there is a solution improvement then
 Intensification phase ();
  Explore randomly the neighbours of the best solution found so far
      (elite solution);
 Global Pheromone update phase ();
  Update pheromone trails for assignments appearing in solution;
else
 Diversification phase ();
  Pheromone evaporation; // diversity control
  Reinitialize pheromone trails;
  Generate new population of ant solutions using elite solutions in
      the external memory by performing some perturbations;
end if
end while
Return best ant // best solution
```

Fig. 1 The pseudo code of the hybrid Elitist-AS for course timetabling problem [15].

Each ant presents a solution which will be improved using the ILS (as in [5]) for a significant enhancement of its quality. Once an elite solution is found, it will be stored in the external memory. This solution will be utilized in the successive iterations as a reference to guide the search toward the global solution. If there is any improvement made to a solution, the intensification phase will proceed to explore furthermore the neighbours of the solution in order to generate new elite solution. If no improvement made for a predefined number of iterations (stagnation state), the intensification phase will be skipped and the diversification phase is commenced. The diversification phase will reinitialize the pheromone trail values to trigger the search again. The whole steps will be repeated until the stopping criterion is met, which is either the maximum number of iterations or a global (lower bound) solution is found.

## 3.2 Hybrid Big Bang-Big Crunch

The hybrid BB-BC (the BB-BC was originally proposed by [23]) is basically a search algorithm which is inspired by the theory of the universe evolution (life cycle). This approach is mainly characterized by a fast search space exploration and aggressive solution space exploitation [24]. This is presented by a population size reduction. The pseudo code of our hybrid BB-BC is illustrated in Fig. 2 [16].

```
Big Bang phase (solutions construction):
Generate population (construct solutions from scratch for the first
        generation, or else generate new population from the
        elite pool) & measure Euclidean distances among
        solutions in the population;
Big Crunch phase (Local Search move):
Repeat
    Generate some neighbors for all solutions in the population and
        replace the parent with its best off-spring for each solution
        in the population;
    Find the centre of mass; //best solution found so far
    Apply local search to the centre of mass;
    Update the elite pool and the best found solution;
    Eliminate some poor quality solutions;
Until population size is reduced to a single solution;
Return to Big Bang phase if stopping criterion is not met;
Return the best found solution
```

Fig. 2 The pseudo code of the hybrid BB-BC for course timetabling problem [16].

The hybrid BB-BC starts the search with the same constructive heuristic as used in the hybrid Elitist-AS approach. A large population of initial solution is generated in the Big Bang phase, and the Euclidean distances among solutions are calculated. That is to measure their attractiveness and their diversity toward/from the yet found elite solution (the differences between solutions' fitness values). In the Big Crunch phase, a number of neighbours for all solutions in the population are generated. The parent solutions are replaced by the some good quality off-springs in order to enforce the population converge towards better quality solution. An elite solution (*centre of mass*) is determined based on its quality, which is the best quality cost among solutions in the population. Then, for a significant enhancement of the *centre of mass* quality, a simple descent heuristic (as a local search) is applied to the *centre of mass*. Once a new *centre of mass* is found and improved by the local search, it will be stored into elite pool acting as a reference for the search. That is for the purpose of guiding the search toward better solution quality, we generate new successive population(s) by utilizing those elite solutions (centre of masses). This is achieved by performing some perturbations to the elite solutions. In the Big Crunch phase, the population size will be gradually reduced (every iteration) into a single solution by eliminating poor quality solutions. The successive Big Bang phase will generate new population from the solutions of the elite pool rather than generating them from scratch. The whole process is repeated until the stopping criterion is met.

### 3.3 Hybrid Scatter Search

The hybrid SS (the SS was originally proposed by [25]) specifically performs structured combinations of elite collection of high diversity and high quality solutions contained in a dynamic memory. This elite collection is the key element to converge the search toward good quality solutions while diversifying the search.

As in genetic algorithms, SS concerns with producing a solution from the combination of elements from other two or more solutions to yield better solutions than the original ones. Recently, SS became one of the state-of-the-art methods for designing solution procedures for hard combinatorial optimization problems [26]. The pseudo code of our SS is shown in Fig. 3.

```
Diversification Generation Method;
    Employ constructive heuristic (e.g. largest degree) to generate
        initial population;
Improvement Method;
    Employ Hill climber to enhance the quality of the population;
Repeat
    Reference Set Update Method;
        Maintain diversity of elite solutions using similarity
            measurement and dynamic update;
    Subset Generation Method;
        Employ Type-I selection; // select one solutions from each
            subset in the reference set
    Solution Combination Method;
        Perform one-point crossover;
    Improvement Method;
        Employ Iterated local search routine to enhance the
            quality of combined solutions;
Until (StoppingCriterion);
Return the best solution found
```

Fig. 3 The pseudo code of the hybrid SS for course timetabling problem [17].

The hybrid SS starts the search with the diversification method by generating a small population of initial solutions from scratch using constructive heuristics (e.g. largest degree). The whole population is then improved in the improvement method using a hill climbing procedure. This is intentionally used to direct the search toward the local optima.

In the reference set update method, a reference set (*RefSet*) of elite and diverse solutions is created (for the first iteration) and will be updated due time once a better quality or diverse solution than those in the *RefSet* in produced. The *RefSet* has two subsets: *b1* and *b2*. Elite solutions are selected based on their quality and then stored in *b1*, whilst diverse solutions are selected based on their greatest dissimilarity from others in the population and then stored in *b2*. The solution that has more uncommon assignments from other solutions (course into timeslot), the more diverse it becomes. The *RefSet* is updated by replacing the worst elite solution in b1 by a better newly generated solution. While a worst diverse solution is replaced by a newly generated solution that has much dissimilarity from the ones in the *b2*.

Then the subset generation method is proceeded which selects one solutions from each subset in the *RefSet* to be combined and to generate new promising solutions. Those selected two solutions are from b1 and b2. This selection mechanism is called *Type-I* method [25], which is the combination of all 2-elements subsets. This means, combining all possible unrepeated two solutions.

In our work, the solution combination is performed using a single-point crossover operator to generate two off-springs. Feasibility of the off-springs is ensured by a repair function that rectifies a corrupted solution resulted by the crossover. These off-springs are further enhanced by the improvement method (e.g. the iterated local search). The improved off-springs will be compared to the ones in the *RefSet* for updating its contents. Then, a successive diversification generation method is commenced once again with the same population size. The new population is generated by performing some perturbations to the solutions in *RefSet* rather than building them from scratch. The whole process is repeated until the stopping criterion is met.

## 4. Experiments and Results

In this work, we tested our hybrid approaches on well known benchmark post-enrolment course timetabling instances (TTComp2003 and ITC2007-Track2). We ran our approaches 25 times (for each) on each instance for a restricted running time 474 seconds for the TTComp2003, and 494 seconds for the ITC2007-Track2. The experiments were performed on Intel Pentium Core2 Duo 2.16 GHz processor, 2GB RAM, and implemented in Java NetBeans IDE v 6.9. Parameters shown in Table 1 are determined experimentally (e.g. elite pool size) and based on the literature (e.g. Elitism). For example, the population size in the Elitist-AS and SS is preferred to be relatively small [18] [27], while the BB-BC follows the typical population size as of the genetic algorithms.

Table 1: Parameters settings used by our hybrid algorithms

| **Parameter** | **Description** |
|---|---|
| *Population size* | Elitist-AS (20);<br>BB-BC (100);<br>SS (50). |
| *Stopping criterion* | No. Of iterations =100,000 or<br>time limit is reached |
| *No. of Non-improvement iterations* | Elitist-AS (100);<br>BB-BC (30);<br>SS (30). |
| *Elite pool size* | Elitist-AS (5);<br>BB-BC (10);<br>SS (20). |
| *Similarity measurement* | Elitist-AS (Non);<br>BB-BC (Euclidean distance, minimum distance from centre of mass);<br>SS (Hamming distance, least similar is the best diverse). |
| *Neighborhood structures per solution* | Elitist-AS (5);<br>BB-BC (3);<br>SS (3). |
| *Local search* | Elitist-AS and SS (Iterated local search);<br>BB-BC (Simple Descent heuristic). |
| *Elitism* | Last population solution is forced to be the best |

Tables 2 and 3 show the best results obtained by our approaches based on our parameters presented in Table 1, compared to the best known results obtained by other methodologies (including population-based) applied over the same instances.

Table 2: Results of our hybrid approaches applied to TTComp2003

| Data Set | *Elitist -AS* | *BB-BC* | *SS* | *MM AS* | *EM GD* | *3-SA** |
|---|---|---|---|---|---|---|
| *1* | 56 | 46 | 37 | 65 | 52 | **16** |
| *2* | 18 | 21 | 12 | 36 | 20 | **2** |
| *3* | 60 | 45 | 40 | 69 | 78 | **17** |
| *4* | 75 | 88 | 75 | 138 | 74 | **34** |
| *5* | 81 | 96 | 54 | 143 | 71 | **42** |
| *6* | **0** | **0** | **0** | 24 | 6 | **0** |
| *7* | 3 | 2 | 2 | 24 | 6 | **2** |
| *8* | 3 | 1 | **0** | 28 | 15 | **0** |
| *9* | 23 | 17 | 14 | 36 | 32 | **1** |
| *10* | 70 | 63 | 58 | 75 | 58 | **21** |
| *11* | 39 | 32 | 32 | 50 | 30 | **5** |
| *12* | 91 | 78 | 64 | 95 | 88 | **55** |
| *13* | 66 | 73 | 57 | 79 | 105 | **31** |
| *14* | 22 | 20 | 20 | 73 | 51 | **11** |
| *15* | 28 | 21 | 18 | 31 | 34 | **2** |
| *16* | 8 | 12 | 5 | 23 | 10 | **0** |
| *17* | 100 | 87 | 68 | 108 | 121 | **37** |
| *18* | 28 | 34 | 20 | 26 | 26 | **4** |
| *19* | 59 | 62 | 40 | 108 | 57 | **7** |
| *20* | **0** | **0** | **0** | 5 | 5 | **0** |

Note:
* the best known results obtained so far.
3-SA: an extended work of the official winner [28]; which is a 3-phase Simulated Annealing-based approach;
MMAS: Max-Min Ant System [4];
EMGD: Hybrid of Electromagnetic-Like mechanism with force decay rate Great Deluge [9].

Table 3: Results of our hybrid approaches applied to ITC2007-Track2

| Data Set | *Elitist -AS* | *BB-BC* | *SS* | *ACO* | *GA TS* | *Best know* |
|---|---|---|---|---|---|---|
| *1* | 697 | 541 | 470 | **15** | 523 | **15**[1] |
| *2* | 1025 | 984 | 920 | **0** | 342 | **0**[1] |
| *3* | 194 | 198 | 194 | 391 | 379 | **164**[3] |
| *4* | **219** | 360 | **219** | 239 | 234 | 234 |
| *5* | **0** | **0** | **0** | 34 | **0** | **0**[2] |
| *6* | **0** | **0** | **0** | 87 | **0** | **0**[2] |
| *7* | 8 | 6 | 6 | **0** | **0** | **0**[1] |
| *8* | **0** | **0** | **0** | 4 | **0** | **0**[2] |
| *9* | 1020 | 1067 | 979 | **0** | 1102 | **0**[1] |
| *10* | 364 | 860 | 447 | **0** | 515 | **0**[1] |
| *11* | 293 | 245 | 233 | 547 | 246 | **178**[3] |
| *12* | 227 | **14** | **14** | 32 | 241 | 32[1] |
| *13* | **0** | **0** | **0** | 166 | **0** | **0**[2] |
| *14* | **0** | **0** | **0** | **0** | **0** | **0**[2] |
| *15* | **0** | **0** | **0** | **0** | **0** | **0**[2] |
| *16* | 10 | 1 | 1 | 41 | **0** | **0**[2] |
| *17* | **0** | **0** | **0** | 68 | **0** | **0**[2] |
| *18* | **0** | **0** | **0** | 26 | **0** | **0**[2] |
| *19* | 1770 | 1680 | 1531 | **22** | 121 | **22**[1] |
| *20* | 571 | 563 | 534 | *infeasible* | **304** | **304**[2] |
| *21* | **0** | **0** | **0** | 33 | 36 | **0**[3] |
| *22* | 2383 | 2383 | 2359 | **0** | 1154 | **0**[1] |
| *23* | 1126 | 982 | 982 | 1275 | 963 | **238**[3] |
| *24* | 20 | **3** | **3** | 30 | 274 | 21[3] |

Note:
[1]ACO: Ant Colony Optimization [6];
[2]GATS: Hybrid Genetic Algorithm with Tabu Search [8];
[3]MMA: Combination of Tabu Search & Simulated Annealing with various neighbourhood operators [29]. The official winner.

From Tables 2 and 3 and, Tables 4 and 5 (in Appendix A), the statistical readings of the results obtained by the three hybrid approaches showed that, in many cases, the presented approaches significantly outperformed other approaches (especially the population-based ones) reported in the literature applied on the same benchmark course timetabling instances. Also our hybrid approaches managed to obtain the optimal results (cost =0) for the following instances: 6, 8, and 20 (for TTComp2003); 5, 6, 8, 13, 14, 15, 17, 19, and 21 (for ITC2007-Track2). A number of new best results obtained (so far until the day of submitting this paper) by our hybrid approaches for ITC2007 (Track2) instances are presented in bold, they are: 4 (cost =219), 12 (cost =14), and 24 (cost =3).

Tables 4 and 5 (refer to Appendix A) show the computational statistics of our hybrid approaches, which indicate the performance of our hybrid approaches. It is clear that the hybrid SS is better than the hybrid Elitist-AS and BB-BC approaches it terms of the quality of solutions (see *best*); the consistency of producing feasible and good quality results (see the standard deviation, and the differences between *best* and *median*) across the 25 runs. This is also applied (to some extent) to both Elitist-AS and BB-BC approaches. It is also indicated that the Elitist-AS and the BB-BC are competing each other, as well as they are very close or exactly the same as the SS approach (in some cases).

The hybrid SS outperforms both hybrid Elitist-AS and BB-BC due to the utilization of the elite pool of best quality and best diverse solutions represented by the *RefSet* and performing a combination of two solutions (best quality and best diverse solutions) in the form of one-point crossover. Whilst, the Elitist-AS outperforms the BB-BC due to maintaining a balance between diversification and intensification of the search by employing two mechanism namely the diversification and intensification mechanisms. The BB-BC performed well so far due to manipulating an elite pool of good quality solutions determined every search big bang-big crunch cycle (named as centres of mass), in which it guarantees good quality solutions while maintaining diversity of the search.

## 5. Conclusion

The overall goal of this study was to compare the performance of three hybrid population-based approaches (Elitist-Ant System, Big Bang-Big Crunch, and Scatter Search) for solving the post-enrolment course timetabling problem, by extending their implementation to the 1st and 2nd international timetabling competitions. Generally, good quality solutions are obtained through exploiting an elite pool of good quality to maintain diversity of the search and to generate new good quality population. By utilizing the capabilities of a population-based approach in large search space exploration and best solutions exploitation using neighborhood structures [3], our experimental results indicated that our approaches are able to produce good quality solutions and are competitive (outperformed others and some of the best known results) to many reported results in the literature applied to the competitions' benchmark instances. Generally, the mechanisms and operators employed in our hybrid approaches proved to be significant in the process of enhancing the performance of the approaches. The hybrid approaches were found out effective and efficient in terms of quality and convergence toward the global solution rapidly. In the future, we may investigate some alternative selection and/or recombination mechanisms of elite solutions in our hybrid approaches. That is, to further understand how to maintain a reasonable degree of the search diversity and to achieve an efficient convergence toward a global solution.

### Acknowledgments

The authors wish to thank Ministry of Higher Education (Malaysia) for supporting this work under the Fundamental Research Grant Scheme (FRGS) no. (UKM-TT-02-FRGS121- 2009).

## References

[1] S. Even, A. Itai and A. Shamir, "On the Complexity of Timetable and Multi commodity Flow Problem," SIAM J. Comput. 5: 691-703, 1976.
[2] R. Lewis, "A survey of metaheuristic-based techniques for University Timetabling problems," OR Spectrum. 30: 167-190, 2008.
[3] C. Blum and A. Roli, "Hybrid Metaheuristics: An Introduction, Studies in Computational Intelligence," In: C. Blum, M.J.B. Aguilera, A. Roli, M. Samples (Eds.), Hybrid Metaheuristics: An Emerging Approach to Optimization, Springer-Verlag Berlin, Heidelberg, SCI, 114: 1-30, 2008.
[4] K. Socha, "The Influence of Run-Time Limits on Choosing Ant System Parameters," In: E. Cantu-Paz et al. (Eds.): GECCO 2003, Springer-Verlag, Berlin, Heidelberg, LNCS. 2723: 49-60, 2003.
[5] Rossi-Doria, M. Samples, M. Birattari, M. Chiarandini, M. Dorigo, L. M. Gambardella, J. Knowels, M. Manfrin, M. Mastrolilli, B. Paechter, L. Paquete and T. Stultzle, "A Comparison of the Performance of Different Metaheuristics on the Timetabling Problem," In: Burke, E.K., De Causmaecker, P. (eds.) PATAT 2002. Springer, Heidelberg, LNCS. 2740: 329–354, 2003.
[6] A. Mayer, C. Nothegger, A. Chwatal and G. Raidl, "Solving the post enrolment course timetabling problem by ant colony optimization," In: the Proceeding of the 7th international conference on the practice and theory of automated timetabling (PATAT 2008), 2008.
[7] H. Turabieh, S. Abdullah and B. McCollum, "Electromagnetism-like Mechanism with Force Decay Rate Great Deluge for the Course Timetabling Problem," In: Wen, P., Li, Y., Polkowski, L., Yao, Y., Tsumoto, S., Wang, G. (eds.) RSKT 2009. LNCS, Springer, 5589: 497–504, 2009.
[8] S. N. Jat and S. Yang "A hybrid genetic algorithm and tabu search approach for post enrolment course timetabling," In: the Journal of Scheduling. DOI 10.1007/s10951-010-0202-0. Springer, 2010.
[9] S. Abdullah, H. Turabieh, B. McCollum and P. McMullan, "A hybrid metaheuristic approach to the university course timetabling problem," In: the Journal of Heuristics. DOI 10.1007/s10732-010-9154-y. Springer, 2010c.
[10] E. G. Talbi, "A Taxonomy of Hybrid Metaheuristics," In: the Journal of Heuristics, Kluwer Academic Publishers. Manufactured in The Netherlands. 8: 541-564, 2002.
[11] S. Abdullah, K. Shaker, B. McCollum and P. McMullan, "Dual Sequence Simulated Annealing with Round-Robin Approach for University Course Timetabling," In: Evolutionary Computation in Combinatorial Optimization, Springer Berlin, Heidelberg, LNCS. 6022: 1-10, 2010a.
[12] S. Abdullah, K. Shaker, B. McCollum and P. McMullan, "Incorporating Great Deluge with Kempe Chain Neighbourhood Structure for the Enrolment-Based Course Timetabling Problem," In: J. Yu et al. (Eds.): RSKT 2010, Springer, Heidelberg, LNAI. 6401: 70–77, 2010b.
[13] E. G. Talbi, "Metaheuristics: from design to implementation," Wiley, USA, 2009.

[14] 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,” In: The Journal of Scheduling. 12: 55–89, 2009.
[15] G. Jaradat and M. Ayob, “An Elitist-Ant System for Solving the Post-Enrolment Course Timetabling Problem,” In: Y. Zhang et al. (Eds): DTA/BSBT 2010, Springer, Heidelberg, CCIS. 118: 167-176, 2010a.
[16] G. Jaradat and M. Ayob, “Big Bang-Big Crunch Optimization Algorithm to Solve the Course Timetabling Problem,” In: the Proceedings of the 10th International Conference on Intelligent Systems Design and Applications, IEEE transactions. 1448-1452, 2010b.
[17] G. Jaradat and M. Ayob, “Scatter Search for Solving the Course Timetabling Problem,” In: the Proceedings of the 3rd Conference on Data Mining and Optimization (DMO’11), IEEE transactions. 213-218, 2011.
[18] K. Socha, J. Knowles and M. Samples, “A max-min ant system for the university course timetabling problem,” In: Dorigo, M., Di Caro, G.A., Sampels, M. (eds.) Ant Algorithms 2002. Springer, Heidelberg, LNCS. 2463: 1–13, 2002.
[19] S. Petrovic and E. K. Burke, “University timetabling,” In: Leung, J. (ed.) Handbook of Scheduling: Algorithms, Models and Performance Analysis, ch. 45. CRC Press, Boca Raton, 2004.
[20] The official website of the Metaheursitcis Network: http://www.idsia.ch/Files/ttcomp2002/, Accessed in 20 June 2011.
[21] The official website of the second international timetabling competition: http://www.cs.qub.ac.uk/itc2007/, Accessed in 20 June 2011.
[22] M. Dorigo, V. Maniezzo and A. Colorni, “Positive feedback as a search strategy,” Technical Report 91-016, Dipartimento di Elettronica e Informazione, Policecnico di Milano, Italy, 1991.
[23] K. Erol and I. Eksin, “A new optimization method: Big Bang–Big Crunch. Advances in Engineering Software,” Elsevier, 37: 106–111, 2006.
[24] H. M. Genc and A. K. Hocaoglu, “Bearing-Only Target Tracking Based on Big Bang – Big Crunch Algorithm,” In: The Proceedings of the 3rd International Multi-Conference on Computing in the Global Information Technology (ICCGI '08). DOI 10.1109/ICCGI.2008.53. 229 – 233, 2008.
[25] F. Glover, “Heuristics for integer programming using surrogate constraints,” Decision Sciences. 8: 156–166, 1977.
[26] M. Laguna, “Scatter Search and Path Relinking,” In: M. Ehrgott et al. (Eds.): EMO 2009, Springer-Verlag Berlin, Heidelberg, LNCS. 5467: 1, 2009.
[27] F. Glover, M. Laguna and R. Martí, “Scatter search,” In: Ghosh A, Tsutsui S (eds) Theory and applications of evolutionary computation: recent trends. Springer, Berlin. 519–529, 2002.
[28] P. Kostuch, “The University Course Timetabling Problem with a 3-phase approach,” In: The Practice and Theory of Automated Timetabling V (PATAT 2004), Lecture Notes in Computer Science, E. K. Burke and M. Trick, Eds., vol. 3616. Springer-Verlag. 109–125, 2005.
[29] H. Cambazard, E. Hebrard, B. O´ Sullivan and A. Papadopoulos, “Local search and constraint programming for the post enrolment-based course timetabling problem,” In: the Annals of Operations Research, 1-25, 2010.

# APPENDIX A

Table 4. Computational statistics of our hybrid approaches applied on TTComp2003 instances

| *Instance* | *Hybrid Elitist-Ant System* | | | | *Hybrid Big Bang-Big Crunch* | | | | *Hybrid Scatter Search* | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | *best* | *Std.* | *median* | *worst* | *best* | *Std.* | *median* | *worst* | *best* | *Std.* | *median* | *worst* |
| *1* | 56 | 6.84 | 61 | 84 | 46 | 9.96 | 55 | 78 | 37 | 11.94 | 45 | 78 |
| *2* | 18 | 3.4 | 18 | 31 | 21 | 3.33 | 24 | 31 | 12 | 5.86 | 19 | 31 |
| *3* | 60 | 2.97 | 62 | 70 | 45 | 7.38 | 51 | 70 | 40 | 8.18 | 51 | 70 |
| *4* | 75 | 5.24 | 80 | 96 | 88 | 3.26 | 94 | 101 | 75 | 5.004 | 81 | 94 |
| *5* | 81 | 5.54 | 85 | 96 | 96 | 2.92 | 100 | 109 | 54 | 8.57 | 67 | 86 |
| *6* | **0** | .44 | 0 | 2 | **0** | .37 | 0 | 1 | **0** | .28 | 0 | 1 |
| *7* | 3 | 1.06 | 3 | 6 | **2** | 1.7 | 2 | 9 | **2** | 1.67 | 2 | 8 |
| *8* | 3 | .2 | 3 | 4 | 1 | .74 | 1 | 4 | **0** | .91 | 0 | 4 |
| *9* | 23 | 6.34 | 27 | 45 | 17 | 6.29 | 23 | 39 | 14 | 5.2 | 19 | 32 |
| *10* | 70 | 5.9 | 80 | 90 | 63 | 6.74 | 69 | 89 | 58 | 9.1 | 63 | 89 |
| *11* | 39 | 7.02 | 45 | 63 | 32 | 4.9 | 37 | 50 | 32 | 4.72 | 37 | 46 |
| *12* | 91 | 3.61 | 93 | 104 | 78 | 5.73 | 83 | 98 | 64 | 10.82 | 73 | 99 |
| *13* | 66 | 3.92 | 71 | 81 | 73 | 5.9 | 77 | 94 | 57 | 5.69 | 61 | 73 |
| *14* | 22 | 4.53 | 25 | 40 | 20 | 5.65 | 23 | 36 | 20 | 4.46 | 24 | 35 |
| *15* | 28 | 5.2 | 29 | 47 | 21 | 4.12 | 26 | 36 | 18 | 5.39 | 26 | 37 |
| *16* | 8 | 2.4 | 8 | 17 | 12 | 3.18 | 17 | 21 | 5 | 2.17 | 7 | 11 |
| *17* | 100 | 4.31 | 103 | 115 | 87 | 3.91 | 91 | 100 | 68 | 5.78 | 75 | 86 |
| *18* | 28 | 3.91 | 34 | 43 | 34 | 5.62 | 41 | 52 | 20 | 5.2 | 23 | 35 |
| *19* | 59 | 10.9 | 68 | 94 | 62 | 7.81 | 72 | 92 | 40 | 5.43 | 45 | 56 |
| *20* | **0** | .000 | 0 | 0 | **0** | .2 | 0 | 1 | **0** | .000 | 0 | 0 |

Table 5. Computational statistics of our hybrid approaches applied on ITC2007-Track2 instances

| *Instance* | *Hybrid Elitist-Ant System* | | | | *Hybrid Big Bang-Big Crunch* | | | | *Hybrid Scatter Search* | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | *best* | *Std.* | *median* | *worst* | *best* | *Std.* | *median* | *worst* | *best* | *Std.* | *median* | *worst* |
| *1* | 697 | 56.8 | 706 | 840 | 541 | 99.72 | 706 | 880 | 470 | 129.51 | 706 | 840 |
| *2* | 1025 | 144.1 | 1262 | 1435 | 984 | 160.43 | 1262 | 1400 | 920 | 184.36 | 1262 | 1400 |
| *3* | 194 | 15.04 | 198 | 244 | 198 | 14.4 | 222 | 250 | 194 | 12.75 | 219 | 226 |
| *4* | **219** | 84.8 | 379 | 429 | 360 | 17.9 | 380 | 429 | **219** | 66.68 | 380 | 390 |
| *5* | **0** | 1.3 | 0 | 4 | **0** | 1.2 | 3 | 4 | **0** | .000 | 0 | 0 |
| *6* | **0** | .000 | 0 | 0 | **0** | .000 | 0 | 0 | **0** | .000 | 0 | 0 |
| *7* | 8 | .56 | 8 | 10 | 6 | .6 | 6 | 9 | 6 | .37 | 6 | 7 |
| *8* | **0** | .000 | 0 | 0 | **0** | .000 | 0 | 0 | **0** | .000 | 0 | 0 |
| *9* | 1020 | 64.04 | 1020 | 1227 | 1067 | 82.61 | 1172 | 1303 | 979 | 50.66 | 1020 | 1172 |
| *10* | 364 | 214.44 | 447 | 868 | 860 | 74.69 | 860 | 1234 | 447 | 155.13 | 860 | 868 |
| *11* | 293 | 35.43 | 328 | 393 | 245 | 26.74 | 250 | 338 | 233 | 7.02 | 245 | 250 |
| *12* | 227 | 28.63 | 230 | 298 | **14** | 132.3 | 227 | 378 | **14** | 117.43 | 227 | 298 |
| *13* | **0** | .000 | 0 | 0 | **0** | .000 | 0 | 0 | **0** | .000 | 0 | 0 |
| *14* | **0** | .41 | 1 | 1 | **0** | .88 | 1 | 3 | **0** | .49 | 1 | 1 |
| *15* | **0** | .000 | 0 | 0 | **0** | .000 | 0 | 0 | **0** | .000 | 0 | 0 |
| *16* | 10 | .68 | 10 | 13 | 1 | 4.41 | 10 | 13 | 1 | 4.18 | 10 | 11 |
| *17* | **0** | .88 | 0 | 4 | **0** | 1.25 | 0 | 4 | **0** | .000 | 0 | 0 |
| *18* | **0** | 5.31 | 0 | 13 | **0** | 3.6 | 0 | 13 | **0** | 5.67 | 0 | 13 |
| *19* | 1770 | 45.45 | 1864 | 1914 | 1680 | 88.6 | 1770 | 1955 | 1531 | 133.56 | 1531 | 1864 |
| *20* | 571 | 14.03 | 573 | 612 | 563 | 13.03 | 571 | 612 | 534 | 14.09 | 541 | 571 |
| *21* | **0** | .47 | 0 | 2 | **0** | .7 | 0 | 2 | **0** | .000 | 0 | 0 |
| *22* | 2383 | 23.62 | 2383 | 2453 | 2383 | 22.35 | 2432 | 2449 | 2359 | 23.97 | 2383 | 2435 |
| *23* | 1126 | 192.41 | 1126 | 1608 | 982 | 215.28 | 1454 | 1608 | 982 | 181.82 | 1126 | 982 |
| *24* | 20 | 1.4 | 20 | 27 | **3** | 9.22 | 20 | 31 | **3** | 8.67 | 20 | 3 |