International Journal of Soft Computing 8 (2): 96-107, 2013
ISSN: 1816-9503


# Effect of Elite Pool and Euclidean Distance in Big Bang-Big Crunch Metaheuristic for Post-Enrolment Course Timetabling Problems

Ghaith M. Jaradat and Masri Ayob
Data Mining and Optimization Research Group, Centre of Artificial Intelligence Technology,
Faculty of Information Science and Technology, The National University of Malaysia,
43600 UKM Bangi, Selangor, Malaysia

**Abstract:** In this study, researchers present an investigation of enhancing the capability of the Big Bang-Big Crunch (BB-BC) metaheuristic to strike a balance between diversity and quality of the search. The BB-BC is tested on three post-enrolment course timetabling problems. The BB-BC is derived from one of the evolution theories of the universe in physics and astronomy. The BB-BC theory involves two phases (big bang and big crunch). The big bang phase generates a population of random initial solutions whilst the big crunch phase shrinks those solutions to a single elite solution presented by a centre of mass. The investigation focuses on finding the significance of incorporating an elite pool and controlling the search diversity via the Euclidean distance. Both strategies provide a balanced search of diverse and good quality population. This is achieved by a dynamic changing of the population size, the utilization of elite solutions and a probabilistic selection procedure to generate a diverse collection of promising elite solutions. The investigation is conducted in three stages, first researchers apply the original BB-BC with an iterated local search; second researchers apply the BB-BC with an elite pool and an iterated local search without considering the Euclidean distance and third researchers apply the BB-BC with an elite pool and a simple descent heuristic with utilizing the Euclidean distance. It is found that by incorporating an elite pool without using the Euclidean distance, the BB-BC performs better than the original BB-BC. However, utilizing both elite pool and Euclidean distance have a greater impact on the BB-BC. The third version of BB-BC performs better than both previous versions. Experiments showed that the third version produces high quality solutions and outperforms some approaches reported in the literature.

**Key words:** Big Bang-Big Crunch metaheuristic, elite pool, Euclidean distance, post-enrolment course timetabling problems, Malaysia

## INTRODUCTION

The university course timetabling problem is considered as an NP-hard problem (Even *et al*., 1976) which is difficult to solve for optimality. During the last decade, various metaheuristics have been applied to solve course timetabling problem (Lewis, 2008). Blum and Roli (2008) classified metaheuristics into two classes: population-based and local search metaheuristics. Some common population-based methods applied to the problem are the ant colony optimization (Socha, 2003; Rossi-Doria *et al*., 2003; Mayer *et al*., 2008), memetic algorithm (Turabieh *et al*., 2009; Jat and Yang, 2011) and hybrid evolutionary algorithm (Abdullah *et al*., 2010a).

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 (Talbi, 2002). Whilst, some common local search methods applied to the problem are tabu search (Rossi-Doria *et al*., 2003), simulated annealing (Rossi-Doria *et al*., 2003) and iterated local search (Rossi-Doria *et al*., 2003). The local search 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

**Corresponding Author:** Masri Ayob, Data Mining and Optimization Research Group, Centre of Artificial Intelligence Technology, Faculty of Information Science and Technology, The National University of Malaysia, 43600 UKM Bangi, Selangor, Malaysia

to obtain new ones (Blum and Roli, 2008). In population-based algorithms such as the Big Bang-Big Crunch (BB-BC), elite solutions recombination is performed implicitly (which are move and swap of assignments in a solution representing information exchange between generations of a good quality solution) (Blum and Roli, 2008). The implicit recombination enables the search process to perform a guided sampling of the search space (Blum and Roli, 2008). This recombination technique can effectively find promising areas of the search space (Blum and Roli, 2008).

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 local search metaheuristics such as Talbi (2002, 2009) and Qu *et al*. (2009). 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). In addition, the utilization of an explicit memory (e.g., elite pool), controlling the search diversity and a dynamic manipulation of the population size are also recommended for a better performance of hybrid metaheuristics (Talbi, 2002; Greistorfer, 2000). A good performance is presented by maintaining a balance between diversification and intensification of the search. Therefore, researchers have chosen the BB-BC for the investigational study since, it has a dynamic population size manipulation and a diversity control strategies. The BB-BC lacks only the utilization of a memory (Erol and Eksin, 2006; Jaradat and Ayob, 2010).

In this research, researchers focus on presenting the preliminary investigation of the previous research (Jaradat and Ayob, 2010) conducted to tackle the post-enrolment course timetabling problem (w.r.t. Socha's benchmark datasets). This research mainly aims at illustrating impact of incorporating an elite pool in the BB-BC and the utilization of the Euclidean distance to provide a balance between diversification and intensification of the search. Researchers conclude the performance and consistency of the hybrid BB-BC by testing it on the post-enrolment course timetabling problems.

## MATERIALS AND METHODS

**Description of the problem:** Post-enrolment 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 (Petrovic and Burke, 2004). In this research, researchers test the BB-BC on the benchmark post-enrolment course timetabling instances of three benchmark datasets. They are:

- Metaheuristics Network (TTComp in 2003) including Socha's instances (Socha *et al*., 2002): the original formulation of the problem containing 12 instances
- TTComp in 2003 announced by Metaheuristics Network (TTComp in 2003): the original formulation of the problem containing 20 instances
- ITC2007 (Track2) (Lewis *et al*., 2007): a full formulation of the problem is introduced containing 24 instances considering timeslots availability and precedence and allowing a number of unscheduled courses (aka distance to feasibility)

These benchmark datasets consider only student's preferences. These instances were generated by the Metaheuristic Network (TTComp in 2003). 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 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 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 (Rossi-Doria *et al*., 2003). However, researchers deal only with feasible solutions in the approach. More information about the instances and the problem formulation can be found by Ceschia *et al*. (2011). This study will investigate a population-based metaheuristic to manage a balance between diversification and intensification of the search in order to improve the quality of the timetable. A number of selective comparisons will be made between the results produced in this study and the state-of-the-art reported in the literature.

**The Big Bang-Big Crunch metaheuristic:** There are many other approaches inspired by nature that have not been applied in course timetabling problem such as Big Bang-Big Crunch (BB-BC). The BB-BC has been applied to some combinatorial optimization problems. For example, Erol and Eksin (2006) applied the original BB-BC on truss optimization problem and compared it against GA and an improved GA called Combat-GA (CGA). They showed that the BB-BC had outperformed the CGA in most of the test functions instances in terms of quality and speed. In another research, Kaveh and Talatahari (2009) compared the BB-BC against Particle Swarm Optimization (PSO), Harmony Search (HS) and Ant Colony Optimization (ACO) over the size optimization of space trusses. They showed that the performance of the BB-BC demonstrates superiority over PSO, HS and ACO in computational time and quality of solutions. Lately, the BB-BC was applied to a number of optimization problems such as: target tracking for underwater vehicle detection and tracking (Genc and Hocaoglu, 2008) and engineering optimization (Kripka and Kripka, 2008).

The BB-BC algorithm is based on one of the theories of the universe evolution in physics and astronomy, describing how the universe was created, evolved and how would end. The BB-BC Theory involves two phases: big bang and big crunch. The big bang phase is a set of procedures of energy dissipation in nature term of disordering and randomness. The big crunch phase is a procedure that randomly distributes particles and draws them into an order.

The big bang and big crunch phases represent large search space exploration and best solution exploitation, respectively. The big bang phase (energy dissipation) randomly generates an initial population of feasible solutions (similar to the GA in respect to generating a random initial population).

Populations produced by the big bang phase will be gradually reduced in the big crunch phase. This aims to reduce computational time and to gain quick convergence while maintaining a diversity of solutions. The cost function value of a solution in the population represents a mass. The best solution is represented as the centre of mass (Erol and Eksin, 2006). The centre of mass will attract other solutions. This is because the solutions with greater mass (in the case smaller penalty cost) are potentially much closer to the centre of the search space (universe) or to the point where the big crunch will converge.

Specifically, the BB-BC researchs with a population of variable size (e.g., stellar objects) (Genc and Hocaoglu, 2008). It can maintain the diversity of the search, in which it may avoid being trapped in a local optimum and can converge in a reasonable speed (Kripka and Kripka, 2008). It is like memetic algorithms except it does not involve solutions combination (e.g., crossover) and the mutation is represented by solution's perturbations. The pseudo code of the BB-BC is illustrated in Fig. 1.

Researchers have chosen the BB-BC due to: less parameterized structure, the ability to distribute the search

```
Repeat
  Big Bang phase (solutions construction):
    Generate population and measure the Euclidean distances among the
    population's solutions
  Big Crunch phase (Locals Search move):
    Find centre of mass
    Improve centre of mass
  successive Big Bang phase:
    Generate new limited population around the centre of mass found
      in previous iterations using normal distribution;
      //to transfer the centre of mass to another region
    Decrease the variance of the normal distribution with iteration count
Until (StoppingCriterion); //maximum number of iterations is reached
```

Fig. 1: The pseudo code of the original BB-BC (Erol and Eksin, 2006)

over a various number of solutions instead of one and the capability of quick convergence even in the existence of multiple local minima (Genc and Hocaoglu, 2008). Hopefully this enables the search to rapidly find good quality solutions in different regions in the search space. Generally, the (finalized version) BB-BC algorithm differs from the original BB-BC algorithm (Erol and Eksin, 2006) in terms of representing exploration and exploitation phases (solution construction and improvement). That is, researchers exploit a collection of elite solutions to generate new promising population in successive Big Bang phases where the elite collection contains good quality solutions whereas the original BB-BC rebuilds new solutions from scratch to generate new generation. Researchers employ variable neighbourhood structures and simple descent heuristic (as a local search) whilst Erol and Eksin (2006) examined solution neighbours either using greedy descent or steepest descent. Researchers simply use the quality of the generated solutions and the minimum Euclidean distance to represent the centre of mass (i.e., the best quality solution) and maximum, minimum cost values of solutions in the elite pool (containing local optima solutions) to determine the boundaries (allowable space) of the successive population. Whereas, the original BB-BC calculates solutions' positions (represented by the Euclidean distances and the standard deviation of the population distribution) relatively to the centre of mass in the search space and the magnitude of gravitational attraction that affects the population to converge toward the centre of mass in the Euclidean space (Erol and Eksin, 2006). Originally, the search space boundary is determined by the summation of the Euclidean distances of all solutions in the population. However, researchers also consider measuring the Euclidean distance of the whole population to efficiently control the production of new solutions within a preferable quality limits to converge toward good quality solutions.

The new offsprings for the successive iteration Big Bang phase (and in Big Crunch phase) are normally distributed around the centre of mass ($C_c$) (Erol and Eksin, 2006) (Eq. 1):

$$C_i^{new} = C_c + \sigma \tag{1}$$

Where:

$C_i^{new}$ = The new generated solution i
$\sigma$ = A standard deviation of a normal distribution

The standard deviation decreases as the iterations elapse according to the following formula (Eq. 2) as in (Erol and Eksin, 2006):

$$\sigma = \frac{r\alpha(C_{max} - C_{min})}{k}, \quad 0\frac{\alpha}{k} < 1 \tag{2}$$

Where:

r = A random number between [0,1]
$\alpha$ = A reduction rate of the size of the search space
$C_{max}$ and $C_{min}$ = The upper and lower limits of the elite pool
k = The number of Big Bang phase iterations

Therefore, the new offspring is generated based on (Eq. 1) within the upper and lower bounds. The offsprings are generated by performing some perturbations to the solutions in the elite pool. The lower and upper limits are necessary to control the solutions distribution. In this research, researchers eliminate r by fixing its value to 1, since it has shown no significant impact on the population reduction process in the preliminary experiments.

At the end of the Big Crunch phase (i.e., when a population size is reduced to one solution) a new generation is generated from the elite pool of the previous generations with the same population size (as in the first generation) and starting with the previous centre of mass. That is, the algorithm regenerates a new population from the elite pool (by performing shakings to the solution), bounded by the maximum and minimum of the solutions' cost values of the previous generation (e.g., bounded using Eq. 1).

In order to include possible good quality solutions, researchers allow an extended lower limit. That is, all improved solutions are allowed (although they are not within the boundary) and the upper limit is restricted so as to limit accepting worse solution. The pseudo code of the hybrid BB-BC algorithm is shown in Fig. 2.

**Big Bang phase (solutions construction):**
**Step 1:** Generate population $N_{pop}$ (construct solutions
    from scratch for the first generation
    or else generate new population $N_{newpop}$ from the elite pool)
    and measure Euclidean distances among solutions
    in the population.
**Big Crunch phase (Local Search move):**
**Repeat**
    **Step2:** Generate some neighbours $N_s$ for all solutions in the
        population and replace the parent with its best offspring
        $C_i^{new}$ for each solution $C_i$ in the population.
    **Step 3:** Find the centre of mass $C_i$
    **Step 4:** Apply local search to the centre of mass.
    **Step 5:** Update the elite pool and the best found solution $C_{best}$.
    **Step 6:** Eliminate some poor quality solutions.
**Until** population size is reduced to a single solution.
**Step 7:** Return to step 1 if stopping criterion is not met.
**Step 8:** Return the best found solution.

Fig. 2: The pseudo code of the hybrid BB-BC for course timetabling problem (Jaradat and Ayob, 2010)

The hybrid BB-BC begins with the construction phase (Big Bang) that constructs a population of $N_{pop}$ feasible solutions $C_i$ from scratch (step 1) for the first generation. This phase can also be referred to as the diversification phase. Researchers use a largest degree ordering heuristic with a repair procedure (Landa-Silva and Obit, 2008; Shaker and Abdullah, 2010) to rectify infeasible solutions. Whilst w.r.t. the ITC-2007 instances, researchers use the same ordering (construction) heuristic to satisfy the 4th and 5th hard constraints (where those two hard constraints are much more complicated) but researchers try to leave few unscheduled courses as much as possible. In which once a room is not available in a certain timeslot for a certain course, researchers randomly select a new timeslot (Cambazard *et al*., 2012; Ceschia *et al*., 2011). For the successive Big Bang phase, researchers generate new population from the elite pool (without including the elite solutions themselves in the new population) by performing shaking to solutions in the pool bounded by the upper and lower cost values of solutions in the elite pool.

In the same step (step 1), the Euclidean distances among solutions in the population are measured to establish a diversity control over the search as well as estimating the attractiveness of an elite solution. In which the search diversity might be limited to some extent depending on the differences between solutions' quality values. For example, a difference between two solutions ($C_i$ and $C_{i+1}$) is presented by the (distance d) difference between the fitness values of those solutions ($d(C_i, C_{i+1}) = f(C_i)-f(C_{i+1})$). That is, the greater the difference between $C_i$ and $C_{i+1}$, the greater the probability of solutions to surround each other (gathered in one cluster) in the next iteration. This is considered in order not to diversify the search too much thus converging toward good quality solution (s) effectively and efficiently. In the research, researchers chose the best quality solution with the minimum Euclidean distance, as the centre of mass. A solution with the greater maximum distance is considered as the most diverse solution. Where this solution might has a completely different structure and fitness cost from the elite solutions. The Euclidean distances among solutions in the population (Eq. 3) and the distances between solutions in the population and solutions in the elite pool (Eq. 4) are calculated as follows (Erol and Eksin, 2006; Brownlee, 2011):

$$d_{min}(C_i, C_{i+1}) = \sqrt{\sum_{i=1}^{N}(C_i - C_{i+1})^2} \qquad (3)$$

where, $i \in N_{pop}\{C_1, C_2, ..., C_N\}$

$$d_{min}(p, q) = \sqrt{\sum_{i=1}^{N}(p_i - q_i)^2} \qquad (4)$$

where, $d_{min}$ (p, q) is the distance between each solution (p) in the population and each solution (q) currently in the elite pool (best quality solutions $C_{best}$ including one or more centre of mass $C_c$). For example, a distance between two solutions is expressed as $(f(p_1)-f(p_2))$ where the fitness value (quality) of a solution is subtracted from the other and the distance between a solution and a centre of mass is calculated as $(f(p_i)-f(q_i))$ (Brownlee, 2011). Simply, the Euclidean distance examines the square root of differences between solutions. According to Brownlee (2011) in the nature inspired algorithms, the diversity of the population or the density estimator of the solution space can be measured using the sum of the Euclidean distances between a solution and all other solutions in the population as a measurement of how much that individual contributes to the diversity. A solution with a minimum distance from the elite solution will have greater attractiveness toward that elite solution (centre of mass).

The hybrid BB-BC records the diversity of the population over time where it is calculated as Eq. 3 the minimum average distance of a solution from all other solutions in the population (aka the average distance from all individuals, Bui *et al*., 2007). Whilst with respect to Eq. 4, it is calculated as the minimum distance between a solution in the population and the centre of mass (aka the distance from the best individual of the population, Bui *et al*., 2007). This may avoids getting trapped in local optima (Bui *et al*., 2007).

Then, in step 2, researchers employ the Big Crunch phase (improvement) where first researchers generate some neighbours of all solutions in the population including the centre of mass by performing simple perturbations. This phase can also be referred to as the intensification phase (or a local search move). Each solution (parent) is replaced with the best offspring in order to have better quality solutions in the successive population while maintaining diversity of the search. This is done to avoid premature convergence of the search. That is a diversity of the search is maintained by the keeping some of the poor quality solutions where some poor quality solutions are eliminated from the population that exceeded the upper boundary. The whole cycle of the BB-BC presents the balance between diversity and quality of the search where the big crunch phase (solution space exploitation) shrinks the population gradually into one elite solution whilst the big bang (search space exploration) generates a whole new population of diverse solutions generated from those solutions in the elite pool.

In step 3, researchers determine the centre of mass $C_c$ based on the best solution cost value found ($C_{best}$) and the minimum average distance from the rest of the population. The centre of mass is further improved by applying a simple descent heuristic for a predefined number of non-improvement iterations (step 4). An elite pool (collection) is created/updated in step 5 where the best solutions (centre of mass) of the previous generations are stored in the elite pool and exploited as reference solutions for the Big Bang phase in successive iterations. In this research, researchers use a fixed size of elite pool where in the first iteration; some good solutions were selected to be included in the pool. At each iteration, the elite pool is updated by replacing the worst solution cost in the current centre of mass and solutions. Based on Eq. 1 the search gradually converged to a single solution by reducing the size of the population (step 6). This is done by eliminating poor quality solutions around the centre of mass. The Big Crunch phase is repeated until the population size is reduced to a single solution (singularity).

A new Big Bang phase is commenced when the population size is reduced into a single solution in big crunch phase (step 7) by returning to the first step where a new population is generated from the elite pool. That is, by including the elite solutions in the new population and generating some neighbours from them to form the new population (rather than generating new solutions from scratch as by Erol and Eksin (2006). Researchers include all centre of mass solutions (in the elite pool) in the new population only when the elite pool is fully occupied. That is, to preserve a higher degree of diversity in order to prevent premature convergence. Whilst in early big bangs (where the elite pool is not full yet of centre of mass solutions taken from previous big bangs), researchers exclude any centre of mass (in the elite pool) from being included in the new population. The hybrid BB-BC algorithm's search processes are repeated until the stopping criterion is met which is in the research either the maximum number of iterations is reached or the best quality solution is found. Finally, the BB-BC returns the best found solution (step 8).

In this research, researchers employ three neighbourhood structures at random to the whole population including centre of mass $C_c$ (i.e., in step 1 and 3). At every iteration, researchers generate five feasible neighbours for each solution in $N_{pop}$ and the best neighbthe is chosen to replace its parent solution for the next generation $N_{newpop}$. Researchers employ three neighbourhood structures from Socha (2003) which they are: move a randomly selected course to a random feasible room and timeslot; swap timeslots and rooms of two randomly selected courses and swap all courses of two randomly selected timeslots and rooms.

A simple descent heuristic local search is employed (as a significant intensification mechanism) in order to improve the quality of solutions by exploring their neighbourhoods without sacrificing the diversity of the search. A simple exploration of some neighbourhoods of a solution (e.g., simple shaking such as move a course into a random and feasible timeslot) is employed in the big crunch phase as it may be sufficient enough to escape the local optima.

## RESULTS AND DISCUSSION

In this research, researchers tested three versions of the BB-BC on three benchmark datasets (i.e., Socha, TTComp in 2003 and ITC2007-Track2). Researchers run each BB-BC Version 25 times on each instance for a relaxed running time (which is the number of iterations) under the same conditions (e.g., number of iterations, local search, neighbourhood structures). The experiments are performed on Intel Pentium Core2 Duo 2.16 GHz processor, 2 GB 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 BB-BC follows the typical population size as of the genetic algorithms.

In the experiments, researchers first implemented the original BB-BC algorithm (V1) as by Erol and Eksin (2006) to identify its limitation such as:

- How researchers may utilize the Euclidean distance measurement in controlling the diversity of the search properly
- How researchers may update the search process and manipulate the population cardinality properly (e.g., what kind of solutions researchers need to keep or discard)
- How to utilize the rapid convergence of the algorithm

Then, researchers implement a slightly modified version of the original one (V2). The modifications of V2 are: ignoring the use of the Euclidean distance and gradually reducing the population size very slightly by eliminating only one poor (the very worst) quality solution. Finally, researchers overcome the limitations of the algorithm in V2 by:

Table 1: Parameters settings used by the BB-BC algorithms

| Parameters | Description |
|---|---|
| $N_{pop}$ | Population size = 100 (feasible solutions) |
| k | Number of iterations = 100,000 |
| $k_{ls}$ | Number of non-improvement iteration = 30 |
| $\alpha$ | Reduction rate = 0.8 |
| Elite_pool | Elite pool size = 10 |
| $N_s$ | Number of neighbours created in each generation = 5 |
| Elitism | Last population solution is forced to be always the best |

- Utilizing the Euclidean distance measurement
- A variable population reduction rate
- A local search routine (i.e., simple descent heuristic)
- Utilizing a memory of elite solutions in the subsequent implementation (V3)

The best results obtained by the three versions are illustrated in Table 2 (w.r.t. Socha datasets) as V1-V3, respectively which are also compared to the best known results obtained by other methodologies (including population-based) applied to the same instances. The results reported in the table are all feasible. The differences between the three versions are as follows:

**V1:** It has no elite pool. This makes it unable to establish a reliable and sufficient exchanging of the search experience among the Big Bang and Big Crunch phases. It has also the reduction rate of 10% of the population size which is also not sufficient enough to a better search convergence. By Erol and Eksin (2006), the reduction rate equals to 1 which is not suitable to the problem's nature where the convergence will be incredibly fast without a significant solution's improvement. The diversity of the search and the centre of the mass are measured by calculating the Euclidean distances of the whole population along with the quality. This is appeared to be hard to estimate its effects on the search in meaningful heuristic information. It employs an iterated local search.

**V2:** It has an elite pool. It has no Euclidean distances measurement, instead, the quality of solutions is considered. The reduction rate of the population size is performed by eliminating the very worst quality solution from the population at every iteration. It employs an iterated local search.

**V3:** It has an elite pool. It has Euclidean distances measurement. The reduction rate of the population size is performed by eliminating poor quality solutions around the centre of mass from the population at every iteration. It employs a simple descent heuristic.

Table 2 demonstrates that V3 has the best performance and consistency in obtaining good quality solutions in most of the runs. This is indicated by maintaining a balance between diversity and quality of the search through the interaction between solutions in the elite pool; the Euclidean distance; the variable population reduction rate; re-initiate a new population and the local search routine. Hence, researchers may conclude that by incorporating an elite pool into the BB-BC has played a major role in enhancing the intensification and the diversification of the search (indicated by better results best of V2 and V3 than V1). In addition, the Euclidean distance measurement has also an impact on the intensification process where in V2 the best results are not significantly better than those of V1 and are outperformed by V3.

In comparison, the results are far from the best known results shown in Table 3 but it has outperform some approaches reported in the table for many instances. Researchers might be unable to outperform the best known results because of the local search routine (simple descent heuristic) or the way researchers explore the neighbours around the elite solution (s) is not sufficient enough for a significant solution improvement. Also, it might be due to the computational time which however was not a significant factor in solving the problem as reported in the literature, since, many researchers were concerned with obtaining a high quality timetable regardless of time. Perhaps, since the BB-BC has the advantage of a rapid convergence (Erol and Eksin, 2006), the population size has to be much larger than the size researchers conducted in order to explore much more possibilities of promising regions in the search space and exploit those regions more intensively.

The results also proved that the hybrid BB-BC algorithm (V3) can consistently produce good quality

Table 2: Results of the BB-BC algorithms applied to Socha's instances

| | V1 | | | | V2 | | | | V3 | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Instances | Best | SD | Median | Worst | Best | SD | Median | Worst | Best | SD | Median | Worst |
| Small1 | 0 | 1.30 | 1 | 1 | 0 | 0.89 | 0 | 2 | 0 | 1.30 | 2 | 4 |
| Small 2 | 0 | 1.20 | 1 | 2 | 0 | 0.89 | 0 | 2 | 0 | 1.11 | 1 | 4 |
| Small 3 | 0 | 1.80 | 2 | 8 | 0 | 0.84 | 1 | 2 | 0 | 1.80 | 2 | 7 |
| Small 4 | 0 | 1.10 | 1 | 2 | 0 | 0.45 | 0 | 1 | 0 | 0.70 | 1 | 2 |
| Small 5 | 0 | 0.74 | 1 | 2 | 0 | 0.45 | 0 | 1 | 0 | 0.70 | 1 | 2 |
| Medium1 | 133 | 16.40 | 159 | 276 | 110 | 3.94 | 114 | 120 | 99 | 7.20 | 108 | 124 |
| Medium2 | 128 | 14.10 | 142 | 238 | 108 | 8.52 | 115 | 128 | 102 | 9.72 | 116 | 134 |
| Medium3 | 184 | 15.80 | 260 | 314 | 160 | 16.36 | 169 | 198 | 158 | 19.52 | 183 | 219 |
| Medium4 | 119 | 15.60 | 135 | 201 | 102 | 14.61 | 119 | 140 | 86 | 11.01 | 106 | 129 |
| Medium5 | 108 | 14.20 | 122 | 193 | 76 | 16.63 | 95 | 121 | 79 | 12.24 | 97 | 120 |
| Large | 786 | 65.50 | 828 | 865 | 785 | 62.73 | 830 | 935 | 768 | 45.96 | 837 | 922 |

SD: The Standard Deviation of 25 runs

Table 3: Results of V3 compared to similar approaches applied to Socha's instances

| | | Population-based approaches | | | | | | | Other approaches | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Instances | BB-BC (V3) | MMAS | GASD | EMGD | EGSGA | TMA | FSI | HBMO | RRM | DSA | SA |
| Small1 | 0 | 1.0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Small 2 | 0 | 3.0 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Small 3 | 0 | 1.0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Small 4 | 0 | 1.0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Small 5 | 0 | 0.0 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Medium1 | 99 | 195.0 | 254 | 96 | 139 | 50 | 45 | 75 | 117 | 93 | 9 |
| Medium2 | 102 | 184.0 | 285 | 96 | 92 | 70 | 40 | 88 | 108 | 98 | 15 |
| Medium3 | 158 | 248.0 | 251 | 135 | 122 | 102 | 61 | 129 | 135 | 149 | 36 |
| Medium4 | 86 | 164.5 | 321 | 79 | 98 | 32 | 35 | 74 | 75 | 103 | 12 |
| Medium5 | 79 | 219.5 | 276 | 87 | 116 | 61 | 49 | 64 | 160 | 98 | 3 |
| Large | 768 | 851.5 | 1027 | 683 | 615 | 653 | 407 | 523 | 589 | 680 | 208 |

MMAS: Max-Min Ant System (Socha *et al*., 2002), results shown are the average values; GASD: Hybrid of Genetic Algorithm and Steepest Descent (Abdullah and Turabieh, 2008); GSGA: Extended Guided Search Genetic Algorithm (Jat and Yang, 2011); EMGD: Hybrid of Electromagnetic-like Mechanism with force decay rate Great Deluge (Turabieh *et al*., 2009); TSMA: Incorporation of Tabu Search into Memetic Algorithm (Turabieh and Abdullah, 2009); FSI: Fish Swarm Intelligence (Turabieh *et al*., 2010); HBMO: Honey Bee-Mating Optimization (Sabar *et al*., 2011); RRM: Controlling multi algorithms (Simulated Annealing, Great deluge, Hill Climbing) using Round Robin (Shaker and Abdullah, 2010); DSA: Dual Simulated Annealing (Abdullah *et al*., 2010b); the best known results are recently obtained by a Simulated Annealing approach (Ceschia *et al*., 2011)

timetables (refer to SD and median in Table 2) which are rather comparable to some results obtained by other population-based metaheuristics reported in the literature (indicated by a small difference between best and median and worst where the smaller the difference the more consistent the algorithm). For example, the algorithm has obtained the best results for all small datasets (same as other approaches) and outperformed some population-based metaheuristics for the medium and large sized datasets. This might be due to the following factors:

- The population size reduction may help the search to converge to local minima (centre of mass) in the big crunch phase, whilst regenerating new population in a new big bang phase may help to diversify the search
- Searching some neighbours within the search space boundaries in the big crunch phase may likely to guarantee a significant solution improvement
- Exploiting an elite pool in generating new promising population in successive big bang phases in which it transfers good information about elite solutions to next generations in order to perform a recombination of good quality solutions

The effectiveness of V3 is due to an efficient search space exploration and effective solution space exploitation. This presents the ability of the BB-BC to maintain a balance between the diversity and quality of the search.

An efficient exploration can be achieved by utilizing an elite pool of good quality solutions to prevent premature convergence of the search (Greistorfer, 2000). Solutions (centres of mass) in the elite pool are utilized by the big bang phase to regenerate new solutions (rather than generating from scratch). The big bang phase reinitializes the search with new population to trigger the search again towards the global solution. The elite pool may provide useful information about the location/structure of the global solution. The exploitation of the elite solution (centre of mass) is performed by the simple descent heuristic and the population reduction rate. This is achieved by performing further exploration of an elite solution's neighbours to enhance the quality of the solution significantly rather than just depending on reaching singularity.

However, compared to the state of the art approaches (e.g., simulated annealing presented in Table 3) there are some drawbacks in the BB-BC. First, there is no effective utilization of the solutions (in the elite pool) in order to perform an explicit solutions recombination. This is due to the fact that there is no combination operator (e.g., crossover) to intensify the search around an elite solution. The intensification of an elite solution (in the BB-BC) is utilized in the form of a simple decent heuristic (a local search routine). In which the search converges toward a local minimum in the same neighbourhood of the solution space, whilst a combination operator converges the search toward a local minimum (perhaps) in a different neighbourhood of the solution space. Second, the diversity control is not properly achieved because of the Euclidean distance measurement provides only the location or quality differences between two or more solutions. Therefore, a structured solution recombination is not performed properly in the BB-BC.

In the BB-BC, it is still hard to manipulate in a way suitable for adaptively manipulate the population size during the search with regards to diversity. This is not easily achieved using the Euclidean distance. In other words, the Euclidean distance is unable to count the number of (dis) similar assignments of courses into

Table 4: Results of V3 applied to TTComp2003 instances

| | Population-based | | | | | Local search | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Instances | V3 | MMAS | AS | EMGD | EGSGA | 3-SAx | GD | LS | HA | SA |
| Comp01 | 46 | 65 | 55 | 52 | 54 | 16 | 85 | 63 | 45 | 45 |
| Comp02 | 21 | 36 | 43 | 20 | 25 | 2 | 42 | 46 | 14 | 20 |
| Comp03 | 45 | 69 | 61 | 78 | 44 | 17 | 84 | 96 | 45 | 43 |
| Comp04 | 88 | 138 | 134 | 74 | 132 | 34 | 119 | 166 | 71 | 87 |
| Comp05 | 96 | 143 | 134 | 71 | 97 | 42 | 77 | 203 | 59 | 71 |
| Comp06 | 0 | 24 | 32 | 6 | 3 | 0 | 6 | 92 | 1 | 2 |
| Comp07 | 2 | 24 | 52 | 6 | 12 | 2 | 12 | 118 | 3 | 2 |
| Comp08 | 1 | 28 | 48 | 15 | 23 | 0 | 32 | 66 | 1 | 9 |
| Comp09 | 17 | 36 | 39 | 32 | 21 | 1 | 184 | 51 | 8 | 15 |
| Comp10 | 63 | 75 | 77 | 58 | 53 | 21 | 90 | 81 | 52 | 41 |
| Comp11 | 32 | 50 | 39 | 30 | 46 | 5 | 73 | 65 | 30 | 24 |
| Comp12 | 78 | 95 | 102 | 88 | 96 | 55 | 79 | 119 | 75 | 62 |
| Comp13 | 73 | 79 | 94 | 105 | 69 | 31 | 91 | 160 | 55 | 59 |
| Comp14 | 20 | 73 | 109 | 51 | 13 | 11 | 36 | 197 | 18 | 21 |
| Comp15 | 21 | 31 | 47 | 34 | 35 | 2 | 27 | 114 | 8 | 6 |
| Comp16 | 12 | 23 | 26 | 10 | 12 | 0 | 300 | 38 | 5 | 6 |
| Comp17 | 87 | 108 | 78 | 121 | 104 | 37 | 79 | 212 | 46 | 42 |
| Comp18 | 34 | 26 | 35 | 26 | 39 | 4 | 39 | 40 | 24 | 11 |
| Comp19 | 62 | 108 | 119 | 57 | 63 | 7 | 86 | 185 | 33 | 56 |
| Comp20 | 0 | 5 | 19 | 5 | 2 | 0 | 0 | 17 | 0 | 0 |

MMAS: Max-Min Ant System (Socha *et al.*, 2002); AS: Ant System (Mayer *et al.*, 2008); EMGD: Hybrid of Electromagnetic-like mechanism with force decay rate Great Deluge (Turabieh *et al.*, 2009); EGSGA: Extended Guided Search Genetic Algorithm (Jat and Yang, 2011); 3-SAx: Extended work of the official winner with some refinements and greater number of iterations equals to 1 million (Kostuch, 2005); GD: Great Deluge (Burke *et al.*, 2003); LS: Local Search (Di Gaspero and Schaerf, 2006); HA: Hybrid metaheuristic, a mixture of constructive heuristics (including local search and Tabu Search), Variable Neighbourhood Descent and Simulated Annealing (Chiarandini *et al.*, 2006); SA: Simulated Annealing approach (Ceschia *et al.*, 2011)

timeslots in two solutions. The Euclidean distance is useful to determine the boundaries and distribution of the search space. Also, the Euclidean distance cannot be replaced by any other distance measurements (e.g., Manhattan distance) in the BB-BC.

However, elite solutions are utilized to generate new promising solutions (rather than from scratch) to restart the search with a new diversified population but close in quality to the current centre of mass. The elite pool provides useful information about the location of the global solution (desired centre of mass) presented by the Euclidean distances between solutions in the population and the centre of mass (s).

Furthermore, researchers carried out further experiments by testing V3 on two more benchmark datasets (i.e., TTComp, 2003 and ITC2007-Track2) to support the hypothesis of the effectiveness, efficiency and consistency of the implementation (V3). Table 4 and 5 show the results obtained by V3 (under competitions' stopping condition, i.e., 474 sec for TTComp2003 and 494 sec for ITC2007) compared to the state of the art approaches.

Notice that across all the 25 runs for each instance of both competitions' datasets Tables 4 and 5 V3 is able to obtain feasible solutions, without marking even a single production of an infeasible solution. From Table 4, it is clear that V3 has obtained high quality results compared to most of the presented approaches. It has obtained high quality solutions that are better than most approaches for 3 instances (e.g., comp06, comp07 and comp20).

In the 2nd competition ITC2007 (Track2), Table 5 shows that V3 has obtained optimality (penalty cost = 0) for one third of the instances (comp5, comp6, comp8, comp13, comp14, comp15, comp17, comp18 and comp21). Other results are comparable to the best known results reported in the literature (e.g., comp24, cost = 3). For the instance comp12, the BB-BC (V3) has obtained a high quality timetable (cost = 14) better than the rest of the approaches reported in the Table 4. In addition, V3 outperformed many approaches for most instances shown in Table 5 especially the population-based ones in terms of quality and feasibility.

In the experiments, researchers demonstrate the effectiveness of incorporating the elite pool and a local search as well as measuring the Euclidean distances in enhancing the original BB-BC. Where the elite pool is exploited to maintain a balance between diversity and quality of the search; while the Euclidean distance helps in the search update process.

The local search adds more significance to the process of improving the quality of the solutions. The t-test values in Table 6 (w.r.t consistency and effectiveness) which demonstrates the effectiveness of V3 where the t-test is carried out with 24 degree of freedom at a 0.05 level of significance.

Table 5: Results of V3 applied to ITC2007 (Track2) instances

| | Local search | | | | | | | Population-based | |
|---|---|---|---|---|---|---|---|---|---|
| Instances | V3 | CTI | HA | LSA | HSAT | TSSA | SA | AS | HGATS |
| Comp1 | 541 | 61 | 1482 | 1861 | 1166 | 571 | 59 | 15 | 523 |
| Comp2 | 984 | 547 | 1635 | inf | 1665 | 993 | 0 | 0 | 342 |
| Comp3 | 198 | 382 | 288 | 272 | 251 | 164 | 148 | 391 | 379 |
| Comp4 | 360 | 529 | 385 | 425 | 424 | 310 | 25 | 239 | 234 |
| Comp5 | 0 | 5 | 559 | 8 | 47 | 5 | 0 | 34 | 0 |
| Comp6 | 0 | 0 | 851 | 28 | 412 | 0 | 0 | 87 | 0 |
| Comp7 | 6 | 0 | 10 | 13 | 6 | 6 | 0 | 0 | 0 |
| Comp8 | 0 | 0 | 0 | 6 | 85 | 0 | 0 | 4 | 0 |
| Comp9 | 1067 | 0 | 1947 | inf | 1819 | 1560 | inf | 0 | 1102 |
| Comp10 | 860 | 0 | 1741 | inf | 2091 | 2163 | inf | 0 | 515 |
| Comp11 | 245 | 548 | 240 | 263 | 288 | 178 | 142 | 547 | 246 |
| Comp12 | 14 | 869 | 475 | 804 | 474 | 146 | 267 | 32 | 241 |
| Comp13 | 0 | 0 | 675 | 285 | 298 | 0 | 1 | 166 | 0 |
| Comp14 | 0 | 0 | 864 | 110 | 127 | 1 | 0 | 0 | 0 |
| Comp15 | 0 | 379 | 0 | 5 | 108 | 0 | 0 | 0 | 0 |
| Comp16 | 1 | inf | 1 | 132 | 138 | 2 | 0 | 41 | 0 |
| Comp17 | 0 | 1 | 5 | 72 | 0 | 0 | 0 | 68 | 0 |
| Comp18 | 0 | 0 | 3 | 70 | 25 | 0 | 0 | 26 | 0 |
| Comp19 | 1680 | inf | 1868 | inf | 2146 | 1824 | inf | 22 | 121 |
| Comp20 | 563 | 1215 | 596 | 878 | 625 | 445 | 543 | inf | 304 |
| Comp21 | 0 | 0 | 602 | 40 | 308 | 0 | inf | 33 | 36 |
| Comp22 | 2383 | 0 | 1364 | 889 | inf | 29 | 5 | 0 | 1154 |
| Comp23 | 982 | 430 | 688 | 436 | 3101 | 238 | inf | 1275 | 963 |
| Comp24 | 3 | 720 | 822 | 372 | 841 | 21 | 0 | 30 | 274 |

AS: Ant System (Mayer *et al.*, 2008); HGATS: Hybrid Genetic Algorithm with Tabu Search (Jat and Yang, 2011); CTI: Combination of general purpose Constraint Satisfaction Solver, Tabu Search and Iterated Local Search (Atsuta *et al.*, 2008); HA: Combination of constructive procedure to achieve feasibility and a simulated annealing (Chiarandini *et al.*, 2008); LSA: Local Search Algorithm taken from the Constraint Solver Library combined with Variable Neighbourhood Search algorithms (Muller, 2008); TSSA: Combination of Tabu Search and Simulated Annealing in conjunction with various neighbourhood operators (Cambazard *et al.*, 2012). The official winner; HSAT: Hybrid approach which combines a constructive heuristic, two separate phases of Simulated Annealing and Neighbourhood operators and it is time dependent (Lewis, 2012); SA: Simulated Annealing approach (Ceschia *et al.*, 2011); inf: no feasible solution was obtained with distance to feasibility (DF>0)

Table 6: t-test of V3 for the three benchmark datasets

| Socha | | | TTComp2003 | | | ITC2007 (Track2) | | |
|---|---|---|---|---|---|---|---|---|
| Instances | t-test | p-value | Instance | t-test | p-value | Instance | t-test | p-value |
| Small | 16.03 | 0 | Comp01 | 29.273 | 0.000 | Comp1 | 37.321 | 0.000 |
| Small 2 | 4.843 | 0 | Comp02 | 37.247 | 0.000 | Comp2 | 37.714 | 0.000 |
| Small3 | 6.011 | 0 | Comp03 | 35.421 | 0.000 | Comp3 | 75.232 | 0.000 |
| Small 4 | 6.549 | 0 | Comp04 | 144.526 | 0.000 | Comp4 | 107.545 | 0.000 |
| Small 5 | 7.688 | 0 | Comp05 | 171.842 | 0.000 | Comp5 | 10.258 | 0.000 |
| Medium1 | 76.621 | 0 | Comp06 | 2.138 | 0.043 | Comp6 | $\alpha$ | - |
| Medium2 | 60.186 | 0 | Comp07 | 9.294 | 0.000 | Comp7 | 51.000 | 0.000 |
| Medium3 | 47.848 | 0 | Comp08 | 8.683 | 0.000 | Comp8 | $\alpha$ | - |
| Medium4 | 48.236 | 0 | Comp09 | 19.118 | 0.000 | Comp9 | 69.721 | 0.000 |
| Medium5 | 40.194 | 0 | Comp10 | 52.148 | 0.000 | Comp10 | 58.608 | 0.000 |
| Large | 90.929 | 0 | Comp11 | 39.311 | 0.000 | Comp11 | 48.267 | 0.000 |
| - | - | - | Comp12 | 73.311 | 0.000 | Comp12 | 5.843 | 0.000 |
| - | - | - | Comp13 | 67.597 | 0.000 | Comp13 | $\alpha$ | - |
| - | - | - | Comp14 | 22.176 | 0.000 | Comp14 | 4.993 | 0.000 |
| - | - | - | Comp15 | 31.749 | 0.000 | Comp15 | $\alpha$ | - |
| - | - | - | Comp16 | 25.823 | 0.000 | Comp16 | 8.851 | 0.000 |
| - | - | - | Comp17 | 117.371 | 0.000 | Comp17 | 2.551 | 0.018 |
| - | - | - | Comp18 | 36.538 | 0.000 | Comp18 | 1.445 | 0.161 |
| - | - | - | Comp19 | 46.104 | 0.000 | Comp19 | 101.804 | 0.000 |
| - | - | - | Comp20 | 1.000 | 0.327 | Comp20 | 220.654 | 0.327 |
| - | - | - | - | - | - | Comp21 | 2.751 | 0.017 |
| - | - | - | - | - | - | Comp22 | 541.400 | 0.000 |
| - | - | - | - | - | - | Comp23 | 30.242 | 0.000 |
| - | - | - | - | - | - | Comp24 | 8.442 | 0.000 |

$\alpha$: t cannot be computed because the standard deviation is 0

## CONCLUSION

The goal of this study was to present the effectiveness of utilizing an elite pool and the Euclidean distance in the BB-BC to enhance its capability of maintaining a balance between the diversification and intensification of the search. Three versions of the BB-BC were tested on a post-enrolment course timetabling problem to support the hypothesis of utilizing an explicit memory and diversity control strategies. The performance of the three versions was relatively good and consistent but still inferior to the best state of the art approaches reported in the literature. An advantage of the hybrid BB-BC is that the capability of exploration and exploitation of multiple solutions. Although, since it has fast solution convergence and it diversifies the search it still has some drawbacks such as the diversity control and performing a structured recombination of elite solutions in order to produce good quality solutions. In the future, researchers may investigate some alternative selection and/or recombination mechanisms of elite solutions in the BB-BC. That is, to further understand how to maintain a reasonable degree of the search diversity and to achieve an efficient convergence toward better quality solutions as well as to achieve better performance and consistency.

## ACKNOWLEDGEMENTS

Researchers thank Ministry of Higher Education for supporting this research under the FRGS Research Grant Scheme (FRGS/1/2012/SG05/UKM/02/11).

## REFERENCES

Abdullah, S. and H. Turabieh, 2008. Generating university course timetable using genetic algorithms and local search. Proceedings of the 3rd International Conference on Convergence and Hybrid Information Technology, Volume 1, November 11-13, 2008, Busan, pp: 254-260.

Abdullah, S., K. Shaker, B. McCollum and P. McMullan, 2010a. Dual sequence simulated annealing with round-robin approach for university course timetabling. Evol. Comput. Comb. Optim., 6022: 1-10.

Abdullah, S., H. Turabieh, B. McCollum and P. McMullan, 2010b. A hybrid metaheuristic approach to the university course timetabling problem. J. Heurist., 18: 1-23.

Atsuta, M., K. Nonobe and T. Ibaraki, 2008. ITC2007 Track 2: An approach using general CSP solver. http://www.cs.qub.ac.uk/itc2007/winner/bestcourse solutions/Atsuta_et_al.pdf.

Blum, C. and A. Roli, 2008. Hybrid Metaheuristics: An Introduction, Studies in Computational Intelligence. In: Hybrid Metaheuristics: An Emerging Approach to Optimization, Blum, C., M.J.B. Aguilera, A. Roli and M. Samples (Eds.). Springer-Verlag Berlin, Heidelberg, Pages: 290.

Brownlee, J., 2011. Clever Algorithms: Nature-Inspired Programming Recipes. 1st Edn., Lulu Press, Australia, ISBN: 9781446785065, Pages: 436.

Bui, L.T., M.H. Nguyen, J. Branke and H.A. Abbass, 2007. Tackling Dynamic Problems with Multiobjective Evolutionary Algorithms. In: Multiobjective Problem Solving from Nature: From Concepts to Applications, Knowles, J., D. Corne and K. Deb (Eds.). Springer, Heidelberg, pp: 77-92.

Burke, E.K., Y. Bykov, J.P. Newall and S. Petrovi, 2003. A time-predefined approach to course timetabling. Yugoslav J. Oper. Res., 13: 139-151.

Cambazard, H., E. Hebrard, B. O'Sullivan and A. Papadopoulos, 2012. Local search and constraint programming for the post enrolment-based course timetabling problem. Ann. Oper. Res., 194: 111-135.

Chiarandini, M., C. Fawcett and H.H. Hoos, 2008. A modular multiphase heuristic solver for post enrollment course timetabling. Proceeding of the 7th International Conference on the Practice and Theory of Automated Timetabling, August 18-22, 2008, Montreal, Canada.

Chiarandini, M., M. Birattari, K. Socha and O. Rossi-Doria, 2006. An effective hybrid algorithm for university course timetabling. J. Sched., 9: 403-432.

Di Gaspero, L. and A. Schaerf, 2006. Neighborhood portfolio approach for local search applied to timetabling problems. J. Math. Model. Algorith., 5: 65-89.

Erol, O.K. and I. Eksin, 2006. A new optimization method: Big Bang-Big Crunch. Adv. Eng. Software, 37: 106-111.

Even, S., and A. Itai and A. Shamir, 1976. On the complexity of timetable and multi commodity flow problem. SIAM J. Comput., 5: 691-703.

Genc, H.M. and A.K. Hocaoglu, 2008. Bearing-only target tracking based on Big Bang- Big Crunch algorithm. The Proceedings of the 3rd International Multi-Conference on Computing in the Global Information Technology, July 27-Augut 1, 2008, Athens, Greece, pp: 229-233.

Greistorfer, P., 2000. On the algorithmic design in heuristic search. Proceedings of the 17th European Conference on Operational Research, July 16-19, 2000, Budapest, Ungarn.

Jaradat, G. and M. Ayob, 2010. Big Bang-Big Crunch optimization algorithm to solve the course timetabling problem. Proceedings of the 10th IEEE International Conference on Intelligent Systems Design and Applications, November 29-December 1, 2010, Cairo, Egypt, pp: 1448-1452.

Jat, S.N. and S. Yang, 2011. A hybrid genetic algorithm and tabu search approach for post enrolment course timetabling. J. Schedul., 14: 617-637.

Kaveh, A. and S. Talatahari, 2009. Size optimization of space trusses using Big Bang-Big Crunch algorithm. Comput. Struct., 87: 1129-1140.

Kostuch, P., 2005. The University Course Timetabling Problem with a 3-Phase Approach. In: The Practice and Theory of Automated Timetabling, Burke, E.K. and M. Trick (Eds.). S pringer-Verlag, Heidelberg, pp: 109-125.

Kripka, M. and R.M.L. Kripka, 2008. Big crunch optimization method. Proceedings of the International Conference on Engineering Optimization, June 1-5, 2008, Rio de Janeiro, Brazil, pp: 1-6.

Landa-Silva, D. and J.H. Obit, 2008. Great deluge with nonlinear decay rate for solving course timetabling problems. Proceedings of the 4th International IEEE Conference on Intelligent Systems, September 6-8, 2008, Varna, Bulgaria, pp: 8.11-8.18.

Lewis, R., 2008. A survey of metaheuristic-based techniques for University Timetabling problems. OR Spectrum, 30: 167-190.

Lewis, R., 2012. A time-dependent metaheuristic algorithm for post enrolment-based course timetabling. J. Ann. Oper. Res., 194: 273-289.

Lewis, R., B. Paechter and B. McCollum, 2007. Post enrolment based course timetabling: A description of the problem model used for track two of the second international timetabling competition. Cardiff Working Papers in Accounting and Finance A2007-3, Cardiff Business School, Cardiff University, Wales.

Mayer, A., C. Nothegger, A. Chwatal and G. Raidl, 2008. Solving the post enrolment course timetabling problem by ant colony optimization. Proceedings of the 7th Ýnternational Conference on the Practice and Theory of Automated Timetabling, August 18-22, 2008, Montreal, Canada, pp: 1-13.

Muller, T., 2008. ITC2007 solver description: A hybrid approach. Proceedings of the 7th International Conference on the Practice and Theory of Automated Timetabling, August 18-22, 2008, Montreal, Canada.

Petrovic, S. and E.K. Burke, 2004. University Timetabling. In: Handbook of Scheduling: Algorithms, Models and Performance Analysis, Leung, J. (Ed.). CRC Press, Boca Raton, USA.

Qu, R., E.K. Burke, B. McCollum, L.T.G. Merlot and S.Y. Lee, 2009. A survey of search methodologies and automated system development for examination timetabling. J. Schedul., 12: 55-89.

Rossi-Doria, O., M. Samples, M. Birattari, M. Chiarandini and M. Dorigo *et al*., 2003. A comparison of the performance of different metaheuristics on the timetabling problem. Proceedings of the Practice and Theory of Automated Timetabling III, August 13-15 2003, Springer, Heidelberg, pp: 329-354.

Sabar, N.R., M. Ayob, G. Kendall and R. Qu, 2011. A honey-bee mating optimization algorithm for educational timetabling problems. Eur. J. Operat. Res., 216: 533-543.

Shaker, K. and S. Abdullah, 2010. Controlling multi algorithms using round robin for university course timetabling problem. Database Theory Appl. Bio-Sci. Bio-Technol., 118: 47-55.

Socha, K., 2003. The ýnfluence of run-time limits on choosing ant system parameters. Proceedings of the Ýnternational Conference on Genetic and Evolutionary Computation, July 12-16, 2003, Springer-Verlag, Berlin, pp: 49-60.

Socha, K., J. Knowles and M. Samples, 2002. A max-min ant system for the university course timetabling problem. Proceedings of the 3rd International Workshop on Ant Algorithms, September 12-14, 2002, Springer, Heidelberg, pp: 1-13.

Talbi, E.G., 2002. A taxonomy of hybrid metaheuristics. J. Heuristics, 8: 541-564.

Talbi, E.G., 2009. Metaheuristics: From Design to Implementation. John Wiley and Sons, New York, USA., ISBN-13: 978-0470278581, Pages: 624.

Turabieh, H. and S. Abdullah, 2009. Incorporating tabu search into memetic approach for enrolment-based course timetabling problems. Proceedings of the 2nd Data Mining and Optimization Conference, October 27-28, 2009, pp: 122 126.

Turabieh, H., S. Abdullah and B. McCollum, 2009. Electromagnetism-like mechanism with force decay rate great deluge for the course timetabling problem. Rough Sets Knowledge Technol., 5589: 497-504.

Turabieh, H., S. Abdullah, B. McCollum and P. McMullan, 2010. Fish swarm intelligent algorithm for the course timetabling problem. Rough Set Knowledge Technol., 6401: 588-595.