Journal of Computer Science

Original Research Paper

An Elite Pool-Based Big Bang-Big Crunch Metaheuristic for Data Clustering

[1]Ibrahim Al-Marashdeh, [2]Ghaith M. Jaradat,
[3]Masri Ayob, [2]Ahmad Abu-Al-Aish and [1]Mutasem Alsmadi

[1]Department of Management Information Systems, College of Applied Studies and Community Service, Imam Abdurrahman Bin Faisal University, Al-Dammam, Saudi Arabia
[2]Department of Computer Science, Faculty of Computer Science and Information Technology, Jerash University, 26150-311 Jerash, Jordan
[3]Data Mining and Optimization Group, Centre of Artificial Intelligence, Faculty of Information Science and Technology, National University of Malaysia, 43600 B. B. Bangi, Selangor, Malaysia

Article history
Received: 24-02-2018
Revised: 23-04-2018
Accepted: 11-06-2018

Corresponding Author:
Ghaith M. Jaradat
Department of Computer Science, Faculty of Computer Science and Information Technology, Jerash University, 26150-311 Jerash, Jordan
Email: ghaith_jaradat@yahoo.com

Abstract: This paper delves into the capacity of enhanced Big Bang-Big Crunch (EBB-BC) metaheuristic to handle data clustering problems. BB-BC is a product of an evolution theory of the universe in physics and astronomy. Two main phases of BB-BC are big bang and big crunch. The big bang phase involves a creation of a population of random initial solutions, while in the big crunch phase these solutions are shrunk into one elite solution exhibited by a mass center. This study looks into enhancing the BB-BC's effectiveness in clustering data. Where, the inclusion of an elite pool alongside implicit solution recombination and local search method, contribute to such enhancement. Such strategies resulted in a balanced search of good quality population that is also diverse. The proposed elite pool-based BB-BC was compared with the original BB-BC and other identical metaheuristics. Fourteen different clustering datasets were used to test BB-BC and the elite pool-based BB-BC showed better performance compared to the original BB-BC. BB-BC was impacted more by the incorporated strategies. The experiments outcomes demonstrate the high quality solutions generated by elite pool-based BB-BC. Its performance in fact supersedes that of identical metaheuristics such as swarm intelligence and evolutionary algorithms.

Keywords: Big Bang-Big Crunch Metaheuristic, Elite Pool, Implicit Recombination, Euclidean Distance, Data Clustering

Introduction

The data clustering problem is classed as NP-hard problem. Gonzalez (1982) mentioned the difficulty in achieving optimal solution for clusters of more than three in number. The last decade has seen the application of numerous metaheuristics in solving numerous data clustering problems (refer to sub-section 3.1). Two classes of metaheuristics as mentioned by Blum and Roli (2008) are: Population-based and local search metaheuristics. Genetic algorithm (Liu *et al*., 2012) and the ant colony optimization (Zhang and Cao, 2011) are among the generally utilised population-based methods in solving the problem. There have been comprehensive investigations on population-based metaheuristics. This type of metaheuristics is popular due to its ability to explore search space exploration, aside from being easily combined with local search methods for improving the process of solution exploitation (Talbi, 2009; Alsmadi, 2016; Alsmadi *et al*., 2012; 2011; Alsmadi, 2017a; 2017b; 2017c; Badawi and Alsmadi, 2014; 2013). Among the general methods of local search methods used on the problem include simulated annealing (Güngör and Ünler, 2007) and tabu search (Liu *et al*., 2008). Their usage is factored by their ability in exploiting the solution space.

Literature Review

Blum and Roli (2008) mentioned the strength of population-based methods being anchored by the ability of recombining solutions in acquiring new ones. Within population-based algorithms for instance the Big Bang-Big Crunch (BB-BC), recombination of elite solutions is implicitly conducted. This entails moving and swapping of assignments within a solution that denote exchange of information between generations of a good quality solution (Blum and Roli, 2008). This refers to the generation of new solutions via a distribution over the search space which comprises a function of previous populations that signify the search experience (Blum and Roli, 2008). Meanwhile, 'implicit' means that a solution is indirectly signified by the assignments' fitness values or their contribution's values to search such as in solution creation. With implicit recombination, Blum and Roli (2008) stated that the process of search could conduct a guided sampling of the search space. Using this recombination technique, potential areas of the search space can be effectively located (Blum and Roli, 2008. The explicit recombination is one more recombination type. It is employed by genetic algorithm, memetic (hybrid genetic) algorithm as well as by scatter search. Here, a structured solution recombination of elite solutions is conducted in an explicit manner. This involves moving or swapping assignments within a solution which denotes exchange of information exchange between generations via one or more recombination operators including mutation and crossover (Blum and Roli, 2008). 'Explicit' means that a solution is directly signified by the actual assignment or the solutions' allocation and fitness values. The selection of the solution recombination is influenced by the nature as well as the construction of the problem and also by the metaheuristic chosen.

Nonetheless, in intensifying the search for solutions of higher quality, the population-based metaheuristic is regarded as weak. As such, specialized metaheuristics in the solution space exploitation (e.g., hill climbing) is generally hybridized with the population-based metaheuristics. This improves the process of intensification. In relation to this, hybridization between a population-based and other local search metaheuristics has been recommended in many studies (Blum and Roli, 2008; Talbi, 2009; Brownlee, 2011). Local search metaheuristics could overcome the shortcoming (in the population-based) of solution space exploitation by improving the quality of solution more (Jaradat *et al*., 2018). Also, to generate better performance of hybrid metaheuristics, the usage of an explicit memory such as the use of elite pool, control on search diversity and dynamically manipulating the population size are also recommended (Talbi, 2009). A good performance can be attained if diversification and intensification of the search stay balanced, which leads to the selection of BB-BC in this study. BB-BC as mentioned by Erol and Eksin (2006) possesses a dynamic population size manipulation and diversity control strategies. The only thing it lacks is a memory usage (Erol and Eksin, 2006).

Elite pool is generally referred as an adaptive memory structure containing a set of diverse and high-quality solutions that keep valuable information about the global optima in the shape of a diverse and elite set of solutions. Using this structure, the process of search could recombine samples from the elite set and this allows the exploitation of valuable information pertaining to the global optima.

Further, to achieve better performance of hybrid metaheuristics, the use of an elite pool of diverse solutions of high-quality for controlling the search in terms of diversity and a dynamic manipulation of the size of the population, are also recommended (Talbi, 2009). As mentioned by Glover *et al*. (2002), a good performance (*w.r.t.* consistency, efficiency, effectiveness and perhaps generality) can be seen via the maintenance of balance between the search's diversification and intensification. This has led to the use of Big Bang Big Crunch (BB-BC) in this study. It comprises hybridization with some mechanisms of diversification and intensification for improving its solution space's exploration and exploitation of the. As demonstrated in the work of Jaradat and Ayob (2013), an elite pool and a local search were used in combination for intensifying the search around elite solutions, with the diversity level maintained.

Objectives

The use of EBB-BC in this study is factored by its: Easy implementation, provision of a deterministic choice of pool of elite solutions both quality and diversity wise which conducts a systematic neighborhood search within the Euclidean space, performance of pseudo-random diversification strategies for the combinations of structured solution, evolution of a renewed strategy via the exploitation of an adaptive memory for the preservation of good quality and diversity, provision of valuable information of elite or diverse solutions even without initial elite pool, support on representation of direct solution within a Euclidean space which can be manipulated easily, capacity in distributing the search over several solutions rather than only one solution as well as the capacity of quick convergence even when multiple local minima is present (Genc and Hocaoglu, 2008) which allows the search to quickly locate the elite solutions within diverse regions, elitism strategy with a pool of only diverse solutions which is enough for the solution space exploration, usage of Euclidean distances for similarities measurement between solutions which assists in pointing the elite solutions and less parameterized structure (Genc and Hocaoglu, 2008) which means freedom from issues of parameter tuning.

BB-BC is also chosen in this study to experiment the impact of using an elite pool together with its recombination of implicit solution. This means that comparison will also be made between this method and others that also employ an explicit recombination. As such, the aim of this study is to investigate the effect of elite pool on the performance of the BB-BC with respect to data clustering problems' solution. With the use of an elite pool, the performance of the BB-BC metaheuristic, in terms of consistency, efficiency, effectiveness as well as generality, is examined by having the method tested on a data clustering problem.

The size of the memory structures in our BB-BC metaheuristic was intentionally fixed in this study. As for the update strategy, it was maintained. Comparison was also made between this method and other similar metaheuristics and standalone methods, including the original BB-BC and particle swarm optimization. The effect of the elite pool in EBB-BC was thus explored in this study.

Therefore, this study attempts to find answer to the research question below:

- Does the usage of elite pool (a pool of diverse and high-quality solutions) combined with an implicit solution recombination improve the performance of BB-BC as opposed to the one that only employs the diverse pool?

As such, this study aims to fulfil two main objectives as follows:

1. To propose an enhanced version of BB-BC via the inclusion of a memory structure (e.g., elite pool) comprising a set of diverse and high-quality solutions in order to achieve balance between diversification and intensification -exploration and exploitation- inside the search space
2. To test the performance of BB-BC in terms of generality and consistency, over a clustering domain with very contrasting characteristics as opposed to combinatorial optimization problems (e.g., course timetabling) and advanced population-based metaheuristics

The arrangement of this paper is as follows: Section 2 highlights the study's problems, section 3 discusses several works pertinent to the subject under study, section 4 illustrates the proposed BB-BC metaheuristic as well as its design, section 5 elaborates the outcomes of the experiment and section 6 concludes the study.

Problem Statement

The subject of data clustering problem has been widely researched and data clustering problem is in fact a very common problem in real life applications. As such, the domain of data clustering offers a very good platform for researcher to test the impact of an elite pool and of other strategies on the performance and generality (consistency and efficiency) of the proposed BB-BC.

As one of the most essential and popular techniques of data analysis, data clustering refers to a process of assembling a set of data objects into clusters. Here, according to Barbakh *et al*. (2009) and Jain (2010), data that belong to the same cluster must be very similar to one another while those belonging to different clusters must be very different from one another.

The evaluation of similarity between data objects usually requires the usage of distance measurement. In particular, the specification of the problem is as follows: Given N objects, each object is allotted to one of K clusters and the sum of squared Euclidean distances between each object and the cluster's centre belonging to each assigned object is minimised:

$$F(O,Z)=\sum_{i=1}^{N}\sum_{j=1}^{K}w_{ij}\left\|\left(O_i-Z_j\right)\right\|^2 \quad (1)$$

Here: $||O_i\text{-}Z_j||$ denotes the Euclidean distance between a data object O_i and the cluster center Z_j. N and K comprise the number of data objects and number of the clusters, respectively. Meanwhile, w_{ij} represents the related weight of data object O_i with cluster j, which will be either 1 or 0 (if object i is allotted to cluster j; w_{ij} is 1, or else, 0). Fuzzy clustering enables w_{ij} to take values in the interval (0, 1).

In this study, BB-BC metaheuristic will be investigated in order to manage a balance between the search's diversification and intensification so that data clustering and analysis will be improved in terms of quality. This study will selectively compare the outcomes of this study with those of the state-of-the-art outcomes documented in the applicable literature.

Related Works

Diverse methodologies have been used for handling different categories of data clustering problems. Thus, the ensuing subsections will highlight some of the most commonly used ones as well as those interesting ones. In should be noted that there has been a wide and successful usage of diverse types of heuristics and metaheuristics for data clustering problems solution. Somehow, the usage of the original BB-BC was only identified once for this purpose.

Data Clustering

The literature has presented countless clustering algorithms. According to Jain (2010), in general, the classical clustering algorithms fall into two categories: Hierarchical algorithms and partitional algorithms. The

author further mentioned that within the domain of classical algorithms, *K*-means is the most recognised algorithm because it is simple and efficient. Somehow, there are two issues that are associated with K-means. First, the number of clusters is required prior to starting that is, the number of clusters must be known a priori. Secondly, as mentioned by Selim and Ismail (1984), the performance of K-means is highly reliant on the initial centroids, aside from its potential in getting stuck in local optima solutions. Thus, within the last 20 years, there have been applications of countless heuristic approaches as an attempt to overcome the problems associated with *K*-means. Among the approaches used include: Simulated annealing by Güngör and Ünler (2007), tabu search by Liu *et al*. (2008), genetic algorithm by Liu *et al*. (2012), neural gas algorithm by Qin and Suganthan (2004), honey bee mating optimization by Fathian *et al*. (2007), artificial bee colony by Karaboga and Ozturk (2011) and Alsmadi (2015), particle swarm optimization algorithm by Kuo *et al*. (2012), ant colony optimization by Zhang and Cao (2011), differential evolution algorithm by Das *et al*. (2009), gravitational search algorithm by Hatamlou *et al*. (2012), firefly algorithm by Senthilnath *et al*. (2011) and Alsmadi (2014), big bang-big crunch algorithm by Hatamlou *et al*. (2011) and black hole heuristic by Hatamlou (2013); all these approaches have been used for data clustering.

Meanwhile, the use of the techniques of clustering can be seen in numerous domains including geophysics (Song *et al*., 2010), agriculture (Chinchuluun *et al*., 2009), image processing (Alsmadi, 2015; 2014; Mitra and Kundu, 2011; Farag *et al*., 2017; Alsmadi, 2017d), document clustering (Cai and Li, 2011), prediction (Chen and Chang, 2010), security and detection of crime (Grubesic, 2006), marketing and costumer analysis (Li *et al*., 2009), anomaly detection (Park *et al*., 2010), medicine (Halberstadt and Douglas, 2008; Abuhamdah, 2015).

Elite Pool

Based on the numerous methods highlighted previously, it can be said that there have been countless efforts of solving the data clustering problems especially via the use of different approaches in combination (hybridization). From all the methods highlighted above, two key properties are salient: (i) First, employ a heuristic method for attaining an initial candidate solution; (ii) second, hybridize the metaheuristic with another heuristic method for improving the solution during the process of iteration. The implementation of primarily population-based hybridization has yielded considerable improvements towards the optimality of the solutions. For instance, population-based methods combined with multiple phase neighborhood search, or greedy randomized adaptive search, or local search, appear to be fairly effective. As stated by Talbi (2009), such hybridization is to expand the strategy of neighborhood in the population-based method.

Further, an adaptive memory structure makes up a key building block of an efficient and effective hybrid metaheuristic, for instance, tabu search algorithms and scatter search. The emphasis is on the notions of memory, intensification versus diversification and exploitation versus exploration. A memory refers to the information gathered by the algorithm on the objective function distribution and is representable as complex structures including trails of pheromone within the Elitist-AS. Meanwhile, intensification exploits the attained information so that the current solutions can be improved. Generally, this entails a local search routine. As for diversification, its aim is to gather fresh information via search space exploration.

These components (e.g., memory, intensification, diversification, elitism, population manipulation and solution recombination) are not always visibly distinctive. They are also very interdependent in an algorithm. As such, in this study, their advantages are used through a complex structure of data that updates the search information in a more effective manner, known as the elite pool. Here, the aim is to fully exploit the adaptive memory; in this study, it is used as an improvement method of the attained best solutions following the combinations.

In the context of the relationships: A pool refers to a data structure employed for keeping several solutions found to be possibly of value all through the search (Greistorfer and Voß, 2005). A pool member is termed an elite solution and thus, elite pool is a notion presentable as an adaptive memory. In relation to this, Rochat and Taillard (1995) made use of the notion of genetic algorithms of combining solutions for the generation of new solutions using a tabu search as a procedure for improvement. Szeto *et al*. (2011) employed the tabu search and unified tabu search. Here, infeasible solutions are considered via the expansion of the objective function using a penalty function and continuous diversification. The approach taken by Mester and Braysy (2007) was similar to (Szeto *et al*., 2011). Also using the elite pool concept, particularly the Granular tabu search, they limited the size of neighbourhood through the removal of edges from the graph that are not likely to emerge in an optimal solution.

All methods highlighted in sub-section 3.1 contain no elite pool of diverse and high quality solutions. Comparatively, the BB-BC proposed in this study contains an incorporated elite pool. Also, these other discussed methods do not employ an implicit solution recombination, unlike the proposed BB-BC. To begin with, the fascinating contributions of the studies mentioned previously, have linkage with the impact of assignment. One way or another, this might impact the performance or even the significance of an elite pool. Owing to their usage on the

same datasets, comparison will be made between some of the methods highlighted in sub-section 3.1 and the proposed BB-BC. The performance of EBB-BC in solving the data clustering problem should be assessed because it would be worthwhile to do so.

The Big Bang-Big Crunch Metaheuristic

Initially introduced by Erol and Eksin (2006), BB-BC is essentially a search algorithm inspired by universe evolution theory which revolves around expansion and shrinking. As described by Genc and Hocaoglu (2008), this algorithm is primarily characterized by a fast search space exploration and aggressive exploitation of solution space. This is signified by shrinking of population in terms of size. The works presented by Erol and Eksin (2006) and Genc and Hocaoglu (2008) provide the details.

This research comprises further investigation on the effect of an elite pool following the inclusion of the performance and generality of the Big Bang-Big Crunch (BB-BC) metaheuristic (from (Jaradat and Ayob, 2013)) by having it tested on datasets of data clustering. Figure 1 which comprises a generic pseudo code of this study's EBB-BC can be referred.

There are many other methods inspired by nature that have been applied to data clustering problems, such as genetic algorithm, *k*-means, particle swarm optimization and gravitational search. The BB-BC has been applied to a limited number of combinatorial optimization problems. For example, Erol and Eksin (2006) applied the original BB-BC to truss optimization problem and compared it against genetic algorithm (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 work, 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; Prayogo *et al*., 2018) and discrete design optimization (Hasançebi and Azad, 2012). Jaradat and Ayob (2013) applied the improved version of the BB-BC to solve course timetabling problems in order to outperform a number of similar methods which showed a consistent and fast convergence towards optimality. The BB-BC has been applied once for the data clustering problem by Hatamlou *et al*. (2011). It showed a good performance as well as generated good quality results.

Numerous other nature inspired methods have been employed for the solution of data clustering problems. These methods include K-means, GA, particle swarm optimization as well as gravitational search. Meanwhile, there has been application of BB-BC to a restricted amount of combinatorial optimization problems. Erol and Eksin (2006) are among those who employed the original BB-BC to the problem of truss optimization and made comparison between this method and GA and an improved GA known as combat-GA (CGA). The outcomes demonstrate that the performance of BB-BC superseded that of CGA in nearly all instances of the test functions with respect to quality as well as speed.

Big Bang phase (solutions construction):
Step 1: Generate population N_{pop} (construct solutions from scratch for the 1st generation, or else generate new population N_{newpop} from elite pool) & measure Euclidean distances among solutions in the population;
Big Crunch phase (Local Search move):
Repeat
Step 2: **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_c;
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. 1: A Generic Pseudo Code of EBB-BC

Further, BB-BC was compared with particle swarm optimization (PSO), harmony search (HS) and ant colony optimization (ACO), in terms of the size optimization of space trusses, in the work by Kaveh and Talatahari (2009). As evidenced, the performance of BB-BC superseded that of PSO, HS and ACO in terms of computational time and solutions quality. BB-BC has also been recently employed in several problems of optimization including target tracking for the detection and tracking of underwater vehicle as can be seen in the work by Genc and Hocaoglu (2008), as well as engineering optimization as shown in the work of Kripka and Kripka (2008). Meanwhile, EBB-BC was used by (Jaradat and Ayob, 2013) in resolving the problems of course timetabling and the method showed better performance when compared with several other identical methods, particularly in terms of consistency and speed of convergence towards optimality. There is one application of BB-BC for the data clustering problem, which is in the work by Hatamlou *et al*. (2011). The authors reported a sound performance and good quality outcomes.

As mentioned, BB-BC is grounded on a theory relating to universe evolution in the realms of physics and astronomy. The theory elucidates the creation, evolution and the ending of the universe. BB-BC theory comprises two phases, namely Big Bang (*BB*) and Big Crunch (*BC*). The *BB* phase comprises a set of procedures of energy dissipation in nature with regard to disordering and randomness while the *BC* phase involves a procedure that arbitrarily dispenses particles and draws these particles into an order.

The phases of *BB* and *BC* both signify large exploration of search space and best exploitation of solution, respectively. The *BB* phase (energy dissipation) involves random creation of an initial population of feasible solutions and this is akin to GA in terms of the creation of a random initial population.

Gradually, the populations generated in the *BB* phase will be reduced in the *BC* phase. Such reduction is for decreasing the computational time and attaining fast convergence, while the solutions' diversity remains the same. The cost function value of a solution within the population signifies a mass and as remarked by Erol and Eksin (2006), the best solution is signified as the *center of mass* which will attract other solutions. Such state is attributable to the notion that solutions with bigger mass (in our context, smaller sum of intra-cluster distances) are possibly much closer to the centre of the search space (the universe), or to the point in which the convergence of the big crunch will occur.

According to Genc and Hocaoglu (2008), BB-BC specifically works with a variable population size for instance, stellar objects. BB-BC can maintain the search diversity. Thus, the problem of being trapped in a local optimum can be prevented while convergence within a reasonable speed can be obtained (Kripka and Kripka, 2008). BB-BC is akin to memetic algorithms but there is no combination of solutions (e.g., crossover) in BB-BC, while the mutation is denoted by perturbations of solution. The summarised comparisons between memetic and BB-BC algorithms are highlighted in Table 1.

In essence, the finalized BB-BC algorithm presented in this study is distinct from the original BB-BC algorithm that (Erol and Eksin, 2006) had introduced, particularly with respect to its representation of exploration and exploitation phases (solution construction and improvement). In particular, an assembly of elite solutions for the creation of new promising population in successive *BB* phases is exploited in this study. Here, the elite collection comprises solutions of good quality. On the other hand, the original BB-BC reconstructs new solutions from scratch in the creation of new generation. Also, variable neighborhood structures and simple descent heuristic (as a local search) are used in this study, On the other hand, Erol and Eksin (2006) scrutinised solution neighbors employing either greedy descent or steepest descent. Additionally, in determining the boundaries (allowable space) of the successive population, this study employs the quality of the produced solutions and the minimum Euclidean distance in representing the *center of mass*, that is, the best quality solution and maximum, minimum cost values of solutions within the elite pool which contains solutions of local optima. Comparatively, in the original BB-BC, the positions of solutions which are denoted by the Euclidean distances and the population distribution's standard deviation are computed relatively to the *center of mass* within the search space and the magnitude of gravitational attraction that impacts the population to converge toward the *center of mass* within the Euclidean space (Erol and Eksin, 2006). The boundary of the search space was initially ascertained using the summation of the Euclidean distances of all solutions within the population. Somehow, to efficiently control new solutions' production within a desirable quality limits for the convergence toward good quality solutions, the measurement of the Euclidean distance of the entire population is also taken into account.

The Euclidean distance assists in the determination of the search space's boundaries and distribution. Actually, in BB-BC, the Euclidean distance is irreplaceable. In other words, no other distance measurements for instance, the Manhattan distance, can be used in this context.

Table 1: A comparison between memetic algorithm and BB-BC

Features	Memetic	BB-BC
Population	Chromosomes, large size	Stellar objects, large size
Reproduction	Probabilistic selection in hamming space	Probabilistic selection in Euclidean space
Combination	Crossover and mutation rely on randomization	No combinations, mutation is a perturbation
Evolution	Survival of the fittest	Strongest gravitational pole
Local Search	Significant intensification	Significant intensification
Search update	Randomized selection	Pseudo- random selection
Memory use	Memory less	Memory less
Search experience	Limited information about solution's components	Limited information about solution's components
Diversification	Mutation	Euclidean distances
Intensification	Selection, Crossover, Replacement	Centre of mass

Normally, the distribution of the new off-springs for the successive iteration *BB* phase as well as in *BC* phase, is around the *center of mass* (C_c) (as in (Erol and Eksin, 2006)) (refer to Equation 2):

$$C_i^{new} = C_c + \sigma \tag{2}$$

Here, C_i^{new} denotes the new produced solution *i*; while σ signifies a standard deviation of a normal distribution. The standard deviation decreases following the elapse of iterations based on the formula below (Equation 3) (Erol and Eksin, 2006):

$$\sigma = \frac{r\alpha(C_{\max} - C_{\min})}{k}, 0 < \frac{\alpha}{k} < 1 \tag{3}$$

Here, *r* represents a random number between [0,1], α denotes a rate of reduction of the search space size, C_{max} and C_{min} represent the elite pool's upper and lower boundaries while *k* represents the number of *BB* phase iterations. As such, the production of the new offspring is according to Equation 2 within the upper and lower limits. The production of off-springs is via the performance of some perturbations to the solutions in the elite pool. It is necessary to have lower and upper boundaries to enable control to the distribution of solutions. In this study, *r* showed no significant impact on the process of population reduction in our initial experiments. Thus, it is taken out by having its value fixed to 1.

At the last part of the *BC* phase signified by the reduction of the population size to one solution, a new generation is created from the earlier generations' elite pool with similar population size (as in the first generation), beginning with the earlier *center of mass*. Here, through shakings performed to the solution, a new population from the elite pool is recreated by the algorithm where the maximum and minimum of the earlier generation's solutions' cost values become the limits (e.g., bounded with Equation 2).

The inclusion of potential good quality solutions is assured through the allowance of an extended lower bound, meaning that, the enhanced solutions are all allowed even those outside the bound, while the upper limit is limited so that the obtainment of worse solution can be limited.

In this study, the proposed BB-BC starts with the construction phase known as the *BB* phase or the diversification phase. This phase comprises the construction of a population of N_{pop} preliminary candidate solutions C_i from scratch (*Step 1*) for the first generation. For the succeeding *BC* phase, new population is created from the elite pool, but the elite solutions themselves are not included in the new population. During this step, shaking is performed to solutions in the pool confined by the upper and lower cost values of solutions within the elite pool.

Also during this step (*Step 1*), measurement is made to the Euclidean distances among solutions within the population. This is for establishing a diversity control over the search and also for estimating an elite solution in terms of its attractiveness. Here, it is possible that the diversity of search is bounded to a certain degree based on the differences between solutions' quality values. As an example, a difference between two solutions namely C_i and C_{i+1} is denoted by the difference of (distance *d*) between the values of fitness those solutions (d (C_i, C_{i+1}) = f (C_i)-f (C_{i+1})). Worded simply, larger difference between C_i and C_{i+1} denotes higher probability of solutions to encircle each other (assembled within one cluster) in the following iteration. Such occurrence is taken into account so that the search is not diversified too much and thus, the convergence is toward solution(s) of good quality effectively as well as efficiently. Solution with the best quality with the minimum Euclidean distance, as the *center of mass* is chosen in this study. The most diverse solution comprises a solution with the larger maximum distance. Such solution may contain structure and fitness cost that are totally different from the elite solutions. The computation of Euclidean distances among solutions in the population as shown in Equation 4, as well as the distances between solutions in the population and solutions in the elite pool as demonstrated in Equation 5 are as follows (Brownlee, 2011; Erol and Eksin, 2006):

$$d_{\min}(C_i, C_{i+1}) = \sqrt{\sum_{i=1}^{N}(C_i - C_{i+1})^2} \tag{4}$$

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

$$d_{\min}(p,q) = \sqrt{\sum_{i=1}^{n}(p_i - q_i)^2} \quad (5)$$

Here: $d_{min}(p, q)$ denotes the distance between each solution (p) in the population and every solution (q) presently in the elite pool (best quality solutions C_{best}, one or more center of mass C_c also included). For instance, a distance between two solutions is stated as ($f(p_1)$-$f(p_2)$), where a solution's fitness value (quality) is subtracted from the other, while the distance between a solution and a center of mass is computed as ($f(p_i)$-$f(q_i)$) (Brownlee, 2011). The Euclidean distance basically looks into the square root of differences between solutions. Brownlee (2011) mentioned that in the nature inspired algorithms, the population diversity or the solution space's density estimator is assessable with the sum of the Euclidean distances between a solution and with the rest of other solutions in the population as an assessment of how much that candidate solution contributes to the diversity. The attractiveness of a solution containing a minimum distance from the elite solution is greater toward that elite solution (*center of mass*).

Over time, the study's proposed BB-BC documents the diversity of the population. The calculation (in terms of Equation 4) comprises the minimum average distance of a solution from all other solutions within the population, which is also termed by Bui *et al.* (2008) as the average distance from all candidate solutions. In terms of Equation 5, the computation comprises the minimum distance between a solution in the population and the *center of mass* which is also termed by Bui *et al.* (2008) as the distance from the best candidate solution of the population. Bui *et al.* (2008) further mentioned that the problem of getting trapped in local optima can also be prevented.

Step 2 involves the *BC* phase (improvement) which is also known as the intensification phase or a local search move. First, several neighbours of all solutions in the population plus the *center of mass* are produced through simple perturbations. The best offspring will replace each solution. This results in better quality solutions in the following population, while diversity of the search remains the same. Such is done so that premature convergence of the search can be prevented, that is, the search diversity is conserved by the retaining some of the poor quality solutions, considering that some of these are taken out from the population that went beyond the upper boundary. The entire BB-BC cycle denotes the balance between diversity and quality of the search. Here, the *BC* phase (solution space exploitation) gradually shrinks the population into a single elite solution. On the other hand, the big bang (search space exploration) produces an entirely new population of diverse solutions from among those within the elite pool.

Step 3 of this study's proposed BB-BC comprises the determination of the *center of mass* C_c according to the discovered best solution cost value (C_{best}) and the minimum average distance from the remainder of the population. The use of a simple descent heuristic for a predefined number of non-improvement iterations (*Step 4*) further improves the *center of mass*. Meanwhile, *Step 5* involves creating and updating an elite pool (collection). Here, the best solutions (*center of mass*) of the earlier generations are kept within the elite pool and used as reference solutions for the *BB* phase in succeeding iterations. Fixed size of elite pool is used in this study; during the first iteration, several good solutions were chosen to be added into the pool. At each iteration the elite pool is updated and this is done through the replacement of the worst solution cost in the present *center of mass* and solutions. As can be seen (Equation 2), reduction of the population size (*Step 6*) leads to a gradual convergence of the search into a single solution. Here, poor quality solutions around the *center of mass* are taken out. The *BC* phase is done over and over until singularity is achieved (i.e., the population size is shrunk to a single solution).

A new *BB* phase starts after the reduction of the population size into a single solution in *BC* phase (*Step 7*). Here, the first step is repeated; a new population is produced from the elite pool via the addition of elite solutions into the new population and the creation of several neighbors from them for the establishment of the new population, instead of creating new solutions from nothing as was laid down by Erol and Eksin (2006). All *center of mass* solutions (in the elite pool) are included in the new population if the elite pool is completely occupied. The purpose of conducting this step is to sustain a higher diversity level so that premature convergence can be avoided. However, in the initial big bangs where the elite pool is yet to be *center of mass* solutions obtained from earlier big bangs, *centres of mass* in the elite pool were all excluded from the new population. The processes of search in the proposed BB-BC algorithm are done over and over until the stopping criterion is satisfied. In other words, the processes will stop when either the maximum number of iterations is achieved, or when the best quality solution is located. Lastly, BB-BC returns the best discovered solution (*Step 8*).

In this study, three neighborhood structures are randomly employed to the entire population *center of mass* C_c included (i.e., in *Step 1* and *Step 3*). Five neighbors are created for every solution in N_{pop} at each iteration. Here, the best neighbor is selected as replacement to its parent solution for the ensuing generation N_{newpop}. The structures of the neighborhood comprise relocating a randomly chosen data object

around one cluster center; swapping two randomly chosen data objects from two randomly chosen cluster centers; and swapping all data objects around two randomly chosen cluster centers.

As substantial mechanism intensification, a simple descent heuristic local search is used. This improves the quality of solutions as their neighborhoods are explored without foregoing the diversity of the search. In the *BC* phase, a simple exploration of several neighborhoods of a solution is used. For instance, simple shaking is performed, such as moving a data object into a randomly chosen cluster center. This may be sufficient in escaping the local optima.

Results

This paper includes the testing of EBB-BC on six benchmark datasets of various complexities. The performance of BB-BC is hence ascertained. The datasets comprise *Iris*, *Wine*, *Glass*, *Wisconsin Breast Cancer*, *Vowel*, *Contraceptive Method Choice* (*CMC*), *Crude Oil* (*CO*), *MG Telescope* (*MGT*), *EGG eye*, *WDBC*, *Ionosphere* (*INS*), *Sonar*, *Thyroid* and *Artset1.* All these datasets are accessible from the repository of the machine learning databases (C.J. Merz, C.L. Blake, UCI Repository of Machine Learning Databases: http://www.ics.uci.edu/-mlearn/MLRepository.html). The key characteristics of the employed datasets are summarised in Table 2.

Experimental Setup

Some researchers including Christofides *et al*. (1979) recommended running every version of BB-BC 25 times on every dataset for 100,000 iterations as a stopping requirement which is a relaxed running time. Intel Core i7 2.30 GHz processor, 8 GB RAM and Java NetBeans IDE v8.1 were employed for the experiments. Parameters are experimentally established (e.g., elite pool size) and is grounded by the literature as well (e.g., Elitism). For instance, in terms of GAs, BB-BC adheres to the classic population size. Table 3 can be referred.

Comparison is made between the proposed BB-BC and the renowned algorithms recently documented in the literature. These include comparison with *K*-means (Jain, 2010), Particle Swarm Optimization (PSO) (Tsai and Kao, 2011), Gravitational Search Algorithm (GSA) (Hatamlou *et al*., 2012), black hole heuristic (BH) (Hatamlou, 2013), Flower Pollination Algorithm (FPA) (Jensi and Jiji, 2015), simplified swarm optimization (SSO) (Yeh and Lai, 2015) as well as big bang–big crunch algorithm (BB-BC) (Hatamlou *et al*., 2011). For this purpose, the Sum of Intra-Cluster Distances (SICD) criteria is used as a measure of internal quality measure: Calculation and summation of the distance between each data object and the center of the corresponding cluster are performed. This is expressed in Equation (1). It is evident that smaller SICD denotes higher quality clustering. In this study, SICD is also the evaluation fitness.

Table 2: Main characteristics of the test datasets

Dataset	Number of clusters	Number of features	Number of data objects
Iris	3	4	150 (50, 50, 50)
Wine	3	13	178 (59, 71, 48)
Glass	6	9	214 (70, 76, 17, 13, 9, 29)
Cancer	2	9	683 (444, 239)
Vowel	6	3	871 (72, 89, 172, 151, 207, 180)
CMC	3	9	1473 (629, 334, 510)
Crude oil	3	5	56
MGT	2	10	19020
EGG eye	2	15	14980
WDBC	2	30	569
Ionosphere	2	34	351
Sonar	2	60	208
Thyroid	3	5	215
Artset1	5	3	250

Table 3: Parameters settings used by our BB-BC algorithm

Parameter	Description
Population size	100
Iterations	100,000
Non-improvement iteration	30
Reduction rate	0.8
Elite pool size	10
Local search routine	Simple descent heuristic
Neighbors created for each generation	5
Search update	Last population solution is forced to be always the best (elitism)

Table 4: The SICD obtained by different algorithms on different datasets compared to our BB-BC

Dataset	Criteria	*K*-means	PSO	GSA	BB-BC	BH	SSO	FPA	Our BB-BC
Iris	*Best*	97.325	96.68	96.679	96.676	96.655	96.66	96.664	96.653
	Avg.	105.729	98.98	96.689	96.765	96.656	96.66	96.673	96.653
	Worst	128.404	127.67	96.705	97.428	96.663	96.66	96.682	96.653
	Std.	12.387	6.96	0.0076	0.2045	0.00173	0.00	0.005	0.00
	T	-	0.75	1.85	-	0.61	0.61	-	0.61
Wine	*Best*	16555.67	16292.22	16294.25	16298.67	16293.41	16292.18	16292.86	16292.04
	Avg.	16963.04	16292.78	16294.31	16303.41	16294.31	16292.76	16293.62	16292.09
	Worst	23755.04	16294.47	16294.64	16310.11	16300.22	16294.17	16294.95	16292.11
	Std.	1180.6942	0.67	0.0406	2.66198	1.65127	0.82	0.636	0.03605
	T	-	18.33	141.91	-	17.64	18.08	-	19.12
Glass	*Best*	215.677	210.36	211.47	223.894	210.515	210.43	211.482	210.365
	Avg.	227.977	217.87	214.22	231.230	211.498	211.31	211.923	211.028
	Worst	260.838	245.32	216.08	243.208	213.956	214.81	214.354	213.861
	Std.	14.13889	7.91	1.1371	4.65013	1.18230	1.78	0.866	1.85685
	T	-	36.94	508.16	-	38.66	39.26	-	34.68
Cancer	*Best*	2986.961	2964.86	2965.14	2964.387	2964.388	2964.39	2964.648	2964.374
	Avg.	3032.247	2966.32	2965.21	2964.387	2964.395	2964.39	2964.994	2964.374
	Worst	5216.089	2969.62	2965.30	2964.389	2964.450	2964.39	2965.445	2964.374
	Std.	315.14560	1.24	0.0670	0.00048	0.00921	0.00	0.235	0.00
	T	-	5.90	11.28	-	5.16	5.39	-	5.84
Vowel	*Best*	149394.803	149089.96	149076.71	149038.516	148985.613	148967.24	-	148076.72
	Avg.	153660.807	151758.39	152289.92	151010.033	149848.181	149148.08	-	149189.49
	Worst	168474.265	17043.64	158612.03	153090.440	153058.986	150139.66	-	150204.63
	Std.	4123.042	4205.04	2947.95	1859.32353	1306.95375	336.16	-	1064.328
	T	-	6.65	13.38	-	6.13	6.44	-	6.07
CMC	*Best*	5542.182	5532.36	5697.03	5534.094	5532.883	5532.18	5534.763	5532.03
	Avg.	5543.423	5532.87	5697.36	5574.751	5533.631	5532.18	5535.529	5532.03
	Worst	5545.333	5533.59	5697.87	5644.702	5534.777	5532.18	5536.020	5532.03
	Std.	1.52384	0.32	0.2717	39.43494	0.59940	0.00	0.363	0.00
	T	-	27.45	54.62	-	26.10	27.12	-	25.51
CO	*Best*	-	277.22	-	-	277.21	277.21	277.251	277.211
	Avg.	-	277.35	-	-	277.27	277.26	277.281	277.211
	Worst	-	277.86	-	-	277.30	277.36	277.313	277.211
	Std.	-	0.13	-	-	0.04	0.05	0.019	0.00
	T	-	1.02	-	-	0.83	0.83	-	0.71
MGT	*Best*	-	1,623322.11	-	-	1,623042.28	1,623042.28	-	1,623042.27
	Avg.	-	1,627770.46	-	-	1,623042.31	1,623045.45	-	1,623042.27
	Worst	-	1,635781.99	-	-	1,623042.38	1,623072.86	-	1,623042.27
	Std.	-	4191.48	-	-	0.03	9.63	-	0.00
	T	-	1211.86	-	-	988.84	1084.09	-	1008.16
EGG eye	*Best*	-	3,010467.48	-	-	2,354713.85	2,354756.19	-	2,354710.15
	Avg.	-	3,210719.24	-	-	2,586299.09	2,354849.12	-	2,354794.64
	Worst	-	3,456696.67	-	-	3,214000.36	2,355129.21	-	2,355036.31
	Std.	-	114894.72	-	-	271990.85	129.73	-	169.2746
	T	-	880.40	-	-	850.98	849.99	-	786.22
WDBC	*Best*	-	149,473.89	-	-	149,473.86	149,473.86	-	149,473.86
	Avg.	-	149,474.13	-	-	149,473.86	149,473.86	-	149,473.86
	Worst	-	149,473.62	-	-	149,473.87	149,473.86	-	149,473.86
	Std.	-	0.20	-	-	0.00	0.00	-	0.00
	T	-	90.58	-	-	90.91	90.76	-	88.13
INS	*Best*	-	793.78	-	-	793.92	793.71	-	793.71
	Avg.	-	793.87	-	-	794.30	793.71	-	793.71
	Worst	-	794.02	-	-	795.34	793.72	-	793.72
	Std.	-	0.07	-	-	0.42	0.00	-	0.00
	T	-	95.97	-	-	105.38	96.35	-	98.48
Sonar	*Best*	-	233.77	-	-	234.22	233.76	-	233.76
	Avg.	-	233.86	-	-	245.02	233.76	-	233.77
	Worst	-	234.08	-	-	266.59	233.77	-	233.78
	Std.	-	0.09	-	-	14.97	0.00	-	0.01
	T	-	301.61	-	-	347.83	328.31	-	323.87
Thyroid	*Best*	-	-	-	-	-	-	1867.862	1867.861
	Avg.	-	-	-	-	-	-	1868.967	1867.861
	Worst	-	-	-	-	-	-	1870.684	1867.861
	Std.	-	-	-	-	-	-	0.926	0.00
	T	-	-	-	-	-	-	-	93.81
Artset1	*Best*	-	-	-	-	-	-	1747.725	1747.18
	Avg.	-	-	-	-	-	-	1747.943	1747.24
	Worst	-	-	-	-	-	-	1748.175	1747.99
	Std.	-	-	-	-	-	-	0.162	0.4513
	T	-	-	-	-	-	-	-	77.23

Experimental Results

Too many instances have been offered for data clustering. Hence, this study decided to test the proposed BB-BC on some customary datasets tested across the literature.

Table 4 presents the summary of the intra-cluster distances attained by clustering algorithms. The documented values include: *best*, *average* (*Avg.*), *worst*, the *standard deviation* (*Std.*) and CPU time (*T*) - in seconds - of solutions over 25 independent simulations. Comparison is made between this study's results and those of the best known algorithms. As can be viewed, best found results are in bold while unfound results are denoted by dashed line.

Table 4 further demonstrate that the results generated by the EBB-BC supersede those of other compared algorithms. Specifically, for the datasets of *Iris*, *Wine*, *Cancer*, *Vowel*, *CMC*, *CO*, *MGT*, *EGG eye*, *Thyroid* and *Artset1*, solutions attained by the EBB-BC are 96.653, 16292.04, 2964.374, 148076.72, 5532.03, 277.211, 1,623042.27, 2,354710.15, 1867.861 and 1747.18, respectively, demonstrating that these solutions are considerably better than those generated by others. For the datasets of *Glass*, *WDBC*, *INS* and *Sonar*, the solutions generated by the EBB-BC are 210.365, 149,473.86, 793.71 and 233.76, respectively; these outcomes are similar to those generated by PSO, BH and SSO. Further, the averages by the EBB-BC are better than those of other algorithms in 12 out of 14 datasets. Also, the values of standard deviation obtained by the EBB-BC are smaller than those of other algorithms in 9 out of 14 datasets. Additionally, worse solutions were obtained by BB-BC for 10 out of 14 datasets; better than the best solutions obtained by the other algorithms. With respect to the time spent in locating the best solution; the EBB-BC is far better than other algorithms in 9 datasets.

In general, as opposed to other best known solutions, the proposed EBB-BC generates high quality solutions and a small standard deviation for every dataset. As opposed to the best known results, the results obtained in this study are either better, or the same, which means that the EBB-BC converges to global optimum in every run, whereas the problem of getting trapped in local optimum solutions may be faced by other algorithms. The EBB-BC did not obtain better average and worst solutions in only the *Vowel* and *Sonar* datasets as opposed to the SSO.

Based on the outcomes obtained, it can thus be said that this study is using a very efficient and competitive methodology in solving the problem of data clustering particularly with respect to solution quality and consistency. As such, the fulfilment of those criteria leads to the generality of this study's proposed BB-BC over diverse sizes of datasets.

Essentially, the EBB-BC proposed in this study has the capacity to employ the ability the heuristic information regarding diverse and high-quality solutions in instance solving, which is through the elite pool, to allow the diversification of the search while intensifying the enhancement of a high-quality solution. As evidenced by the results, the proposed EBB-BC provides a general mechanism irrespective of the nature and complexity of the instances. It is also applicable to other domains with no significant amount of changes to be made prior to the usage. In fact, only the constructive heuristics and neighborhood structures need to be changed. It should be noted that in general, the application of a methodology to other problem areas or even different instances of the same problem necessitate a significant amount of modification, for instance, the modification on algorithm parameters or structures. Comparatively, the EBB-BC can be simply used across different datasets of the clustering problem. It is also hoped that BB-BC would is generalizable to other areas as well.

The performance of the EBB-BC is evaluated using three criteria: generality, consistency and efficiency. Generality refers to the ability the proposed EBB-BC in working soundly across different datasets of the same problem. Meanwhile, consistency refers to the capacity of this algorithm in generating results that are stable when executed a number of times for each dataset. Consistency is generally among the most essential criteria in the evaluation of any algorithm because many search algorithms contains a stochastic component which requires different solutions over multiple runs albeit the same initial solution. The consistency of this study's proposed BB-BC is grounded on the average and the standard deviation over 25 independent runs. Efficiency refers to this algorithm's capacity in generating good results that is almost similar to or superior than the best known value documented in the literature. This study's proposed EBB-BC is measured by reporting, for every dataset, the *Best* and *Avg*. from the best known results documented in the literature.

For each dataset tested, comparison was made between the proposed BB-BC's results with those of identical methods with respect to solution quality instead of computational time. This is because different computer resources employed has made comparison very challenging. As such, the number of iterations being the termination criteria from the usage of the adaptive memory in the proposed EBB-BC was established, resulting in the execution time of this study's proposed algorithm to be within the range of those documented in the literature.

Discussion

This section elaborates the performance assessment of the EBB-BC against other conventional and hybrid algorithms reported in the literature. In

specific, this section will elaborate on: (i) The evaluation of the benefit of integrating an elite pool within the EBB-BC and (ii) the testing on the generality and consistency of the EBB-BC over a problem of data clustering and comparing between the EBB-BC and other well-known algorithms.

For supporting this study's hypothesis on the impact of the elite pool, implicit recombination and Euclidean distance on the performance of BB-BC, this study's EBB-BC is compared with several conventional and hybrid metaheuristics containing no elite pool. As example, a GA usually has a pool (an explicit memory specifically) of diverse solutions, but it has no pool of elite solutions (diverse and high-quality) (Blum and Roli, 2008; Talbi, 2009). This explains why this algorithm possesses a great mechanism of search diversification while lacking efficient intensification mechanism (Blum and Roli, 2008). For certain algorithms including memetic algorithms, honey bee mating and gravitational search algorithms, the usage of elite pool can improve the performance of metaheuristic algorithms in the resolution of various problems of optimization (Resende *et al*., 2010).

A lot of methodologies used in the clustering problem did not include the use of an explicit or implicit memory, which may lead to the lack of sustaining a balance between the search's diversity and quality. A systematic selection strategy is also lacking, making the current study's outcomes outstanding.

The elite pool is structured in a manner that it effectively interacts with the strategy of implicit solution recombination while the Euclidean distance measurement makes available an adaptive search update. Hence, a fairly quick convergence towards high-quality solutions may be certain without having the search diversity sacrificed. As indicated, the EBB-BC has an implicit memory to enable the storage of solutions of high-quality and diverse. Nonetheless, having to directly apply assignments and perturbations can be exhaustive, e.g., apply neighborhood structures that are problem dependent, to good quality or diverse solutions for more quality improvements could be time-consuming.

The effect of the quality of using an elite pool has been determined. Specifically, the conventional BB-BC was applied (Hatamlou *et al*., 2011) with no elite pool. Then, comparison was made to the EBB-BC with an elite pool. Some statistically significant conclusions on the performance of the EBB-BC are worth discussing. Thus, *t*-test was performed out with 24 degree of freedom at a 0.05% significance level. The *p*-value of the EBB-BC as opposed to that of the BB-BC is shown in every criterion particularly the outcomes of *Best* or the *Avg.*, as illustrated in Table 4. As shown, the EBB-BC is statistically better in performance as opposed to the BB-BC in each dataset, with the *p*-value <0.05. The *t*-test values can be viewed in Table 5. The values show the EBB-BC's effectiveness and consistency.

Table 5: *t*-test of the EBB-BC for all datasets

Dataset	*t*-test	*p*-value
Iris	*a*	-
Wine	3.803	0.002
Glass	1.711	0.043
Cancer	*a*	-
Vowel	2.239	0.039
CMC	*a*	-
Crude oil	*a*	-
MGT	*a*	-
EGG eye	4.092	0.001
WDBC	*a*	-
Ionosphere	*a*	-
Sonar	5.145	0.000
Thyroid	*a*	-
Artset1	1.925	0.041

a. t cannot be computed because the standard deviation is 0.

Briefly stated, the obtained outcomes demonstrate the superiority of the EBB-BC with respect to consistency, efficiency and generality, particularly in terms of the tested datasets. This is primarily factored by the usage of elite pool within EBB-BC which imparts a positive impact on the capacity of the EBB-BC in generating outcomes of good quality that are also consistent as opposed to the conventional BB-BC. In all datasets, the *Std*. and the *Avg*. of the EBB-BC shows stable and better outcomes or outcomes that are very close to those generated by other population-based metaheuristic methods. These observations are proofs of the capacity of the EBB-BC in generating good quality outcomes over all datasets, rather than just a few ones.

From the experiments, it is clear that the outstanding performance of the EBB-BC is primarily factored by the hybridization of BB-BC with: An explicit memory structure such as elite pool, an implicit solution recombination and the measurement of Euclidean distance. The purpose is to diversify the search through the exploration of diverse regions of the search space, or rather, by avoiding local optima, while the high-quality solutions are maintained. The result generally shows the significant impact of the hybridization of the elite pool with the BB-BC on its performance in solving the problem of data clustering.

As such, it is clear that the EBB-BC proposed in this study and the conventional BB-BC (Erol and Eksin, 2006; Genc and Hocaoglu, 2008) applied in the work by Hatamlou *et al*. (2011), differ from one another. Firstly, there is no elite pool in the original BB-BC and thus, it is not effective when exchanging search experiences between BB and BC phases. Additionally, having a rate of reduction of 10% in the population size is not enough to attain better convergence; while speed is incredible, there would still be no considerable enhancement. Reduction is conducted by taking out the worst solution from the population at each iteration. Lastly, the original

BB-BC contains Euclidean distances measurement and an iterated local search.

Comparatively, the EBB-BC has both an elite pool and Euclidean distances measurement. With respect to the rate of reduction of the size of population, it is performed by taking out the solutions of poor quality around the *center of mass* from the population at every iteration. The EBB-BC is also a simple descent heuristic.

As can be viewed in Table 4, the EBB-BC shows the best performance and consistency when it comes to acquiring solutions of good quality in nearly all of the runs. Such is evidenced by the maintenance of a balance between the search's diversity and quality via the interaction between solutions in the elite pool, the Euclidean distance, implicit solution recombination, the rate of reduction of the variable population, the restart of a new population as well as the local search routine. It is thus deducible that the inclusion of an elite pool into the BB-BC has majorly contributed to the improvement of the search particularly in terms of intensification and the diversification. Also, the Euclidean distance measurement affects the process of intensification.

As evidenced, the EBB-BC can reliably generate good quality results (see *Std.* and *Avg.*). The results of the EBB-BC are fairly comparable to some of those attained using other metaheuristics documented in the literature (denoted by a small difference between *Best* and *Avg.* and *Worst*, where smaller difference denotes more consistent algorithm). For instance, the results of the proposed EBB-BC are superior as opposed to those from other state-of-the-art metaheuristics for 12 out of 14 datasets.

The superiority of the EBB-BC in terms of the results generated could be linked to some factors. Firstly, reduction on the population size may assist the convergence of the search to local minima or *center of mass* in the phase of *BC.* Meanwhile, the recreation of new population in a new *BB* phase may assist in the diversification of the search. The search of certain neighbors inside the boundaries of the search space in the *BC* phase may likely to assure a considerable improvement to the solution. The EBB-BC includes the exploitation of an elite pool in creating new promising population in succeeding *BB* phases. Here, good information about elite solutions is transferred to next generations so that a recombination of good quality solutions can be performed.

Nonetheless, the usage of elite solutions is for producing new potential solutions (instead of doing it from zero) for restarting the search with new diversified population but with quality almost identical to that of the present *center of mass*. Valuable information is provided by the elite pool particularly in terms of the location of the global solution (the sought after *center of mass*) that is shown by the Euclidean distances between solutions in the population and the *center of mass*(s).

The experiments conducted in this study show the effectiveness of adding the elite pool, a local search and the Euclidean distances, as an attempt to improve the original BB-BC. Here, the elite pool is exploited so that a balance between diversity and quality of the search can be preserved. At the same time, the Euclidean distance and implicit solution recombination provide assistance in the process of search update. With local search, the process of enhancing the solutions' quality becomes more significant.

Conclusion

This study attempted to illustrate the effectiveness of using an elite pool, Euclidean distance and implicit recombination in the BB-BC, in order to improve its ability in keeping a balance between diversification and intensification of the search.

Thus, the effect of an elite pool on the general performance of a population-based metaheuristic was tested. The EBB-BC employs an elite pool containing an assembly of diverse and high-quality solutions. The presence of memory structure assists in preserving a balance between diversity and quality of the search. For instance, escaping local optima, that is, the minima or maxima based on the formulation of a problem; this is doable via the use of new solutions' generation from those diverse ones in the elite pool. The search may be diversified for tapping into new budding domains. Also, it can be converged toward superior quality solutions by having the search focused around good quality solutions from the elite pool.

Testing was conducted on the EBB-BC using a data clustering problem. This was to support the hypothesis of employing an explicit memory and strategies of diversity control. As demonstrated by the results, the EBB-BC generates solutions of high-quality, if not optimal. Also, this algorithm's performance is well generalizable across different datasets or problems. The deduction made by this study is that the hybridization of an elite pool within a population-based metaheuristic can improve its performance that is generalized well across different problems while generating solutions of high-quality that are either competitive or optimal in certain instances.

This study contributes to the reservoir of the applicable domain as highlighted below:

- The creation of the EBB-BC containing an elite pool alongside this algorithm's capacity in conducting heuristic perturbations is a proof that strengths of different search algorithms are combinable into one hybrid methodology. This can be exemplified by constructive heuristics and metaheuristics, as well as population-based and local search methods
- The hybridization of a mechanism of an adaptive memory such as an elite pool containing an assembly of high-quality and diverse solutions, with a

population-based metaheuristic such as BB-BC could yield consistent outcomes that are generalizable across different problem domains or datasets. Also, the proposed algorithm generates high-quality solutions that are just as good as or better than those produced by other comparable methods

- The created hybrid metaheuristic is easily applied to other problem domains with minimal effort. Here, only the constructive heuristics and neighborhood structures require modification
- The usage of an elite pool offers various high-quality solutions from which the proposed EBB-BC initiates the search for obtaining superior solutions. The use of elite pool also offers a way to implement cooperation and attain quicker convergence

The shortcomings of the original BB-BC are generally overcome through the use of: The Euclidean distance measurement, a variable population reduction rate, simple descent heuristic and memory of elite solutions.

This study proposes that the future work investigates the effectiveness of this EBB-BC metaheuristic on other problems, such as big data analytics.

Acknowledgement

The authors wish to thank the Universiti Kebangsaan Malaysia for supporting this work under grant Dana Impak Perdana (DIP-2014- 039).

Author's Contributions

Ibrahim Al-Marashdeh: Participated in all experiments and the design of the Algorithm, coordinated the data-analysis, and contributed to the writing of the manuscript. Designed the research plan and organized the study.

Ghaith M. Jaradat: Participated in all experiments, coordinated the data-analysis, coordinated the design of the Algorithm and contributed to the writing of the manuscript. Designed the research plan and organized the study.

Masri Ayob: Coordinated the data-analysis, coordinated the design of the Algorithm, and contributed to the writing of the manuscript. Designed the research plan and organized the study.

Ahmad Abu-Al-Aish and Mutasem Alsmadi: Participated in all experiments, coordinated the data-analysis and contributed to the writing of the manuscript.

Ethics

To the best of our knowledge this work contains no material previously published by any other person except where due acknowledgement has been made. This work contains no material which has been accepted as part of the requirements of any other academic degree or non-degree program, in English or in any other language.

References

Abuhamdah, A.F., 2015. PLS mechanism for local search algorithm (PPCA) for medical clustering problem. Int. J. Emerg. Sci., 5: 16-37.

Alsmadi, M., 2016. Facial recognition under expression variations. Int. Arab J. Inform. Technol., 13: 133-141.

Alsmadi, M., K. Omar and I. Almarashdeh, 2012. Fish Classification: Fish Classification Using Memetic Algorithms with Back Propagation Classifier. 1st Edn., LAP LAMBERT Academic Publishing, ISBN-10: 3848421674, pp: 180.

Alsmadi, M., K. Omar, S. Noah and I. Almarashdeh, 2011. A hybrid memetic algorithm with back-propagation classifier for fish classification based on robust features extraction from PLGF and shape measurements. Inform. Technol. J., 10: 944-954. DOI: 10.3923/itj.2011.944.954

Alsmadi, M.K., 2014. A hybrid firefly algorithm with fuzzy-C mean algorithm for MRI brain segmentation. Am. J. Applied Sci., 11: 1676-1691. DOI: 10.3844/ajassp.2014.1676.1691

Alsmadi, M.K., 2015. MRI brain segmentation using a hybrid artificial bee colony algorithm with fuzzy-c mean algorithm. J. Applied Sci., 15: 100. DOI: 10.3923/jas.2015.100.109

Alsmadi, M.K., 2017a. An efficient similarity measure for content based image retrieval using memetic algorithm. Egypt. J. Basic Applied Sci., 4: 112-122. DOI: 10.1016/j.ejbas.2017.02.004

Alsmadi, M.K., 2017b. Query-sensitive similarity measure for content-based image retrieval using meta-heuristic algorithm. J. King Saud Univ. Comput. Inform. Sci. DOI: 10.1016/j.jksuci.2017.05.002

Alsmadi, M.K., 2017c. Forecasting river flow in the USA using a hybrid metaheuristic algorithm with back-propagation algorithm. Scientific J. King Faisal Univ., 18: 13-24.

Alsmadi, M.K., 2017d. A hybrid fuzzy c-means and neutrosophic for jaw lesions segmentation. Ain Shams Eng. J.

Badawi, U.A. and M.K. Alsmadi, 2014. A general fish classification methodology using meta-heuristic algorithm with back propagation classifier. J. Theoretical Applied Inform. Technol., 66: 803-812. http://www.jatit.org/volumes/Vol66No3/18Vol66No3.pdfand great deluge local search) with back-propagation classifier for fish recognition. Int. J. Comput. Sci. Issues, 10: 348-356.

Barbakh, W.A., Y. Wu and C. Fyfe, 2009. Review of Clustering Algorithms. In: Non-Standard Parameter.

Badawi, U.A. and M.K.S. Alsmadi, 2013. A hybrid memetic algorithm (genetic algorithm Adaptation for exploratory data analysis. Springer, pp: 7-28.

Blum, C. and A. Roli, 2008. Hybrid Metaheuristics: An Introduction. In: Hybrid Metaheuristics: An Emerging Approach to Optimization, Blum, C., M.J.B. Aguilera, A. Roli and M. Sampels (Eds.), Springer Berlin Heidelberg, pp: 1-30.

Brownlee, J., 2011. Clever Algorithms: Nature-Inspired Programming Recipes. 1st Edn., Jason Brownlee, ISBN-10: 1446785068, pp: 432.

Bui, L.T., M.H. Nguyen, J. Branke and H.A. Abbass, 2008. Tackling Dynamic Problems with Multiobjective Evolutionary Algorithms. In: Multiobjective Problem Solving from Nature, Knowles, J., D. Corne, K. Deb and D.R. Chair (Eds.), Springer, pp: 77-91.

Cai, X. and W. Li, 2011. A spectral analysis approach to document summarization: Clustering and ranking sentences simultaneously. Inform. Sci., 181: 3816-3827. DOI: 10.1016/j.ins.2011.04.052

Chen, S.M. and Y.C. Chang, 2010. Multi-variable fuzzy forecasting based on fuzzy clustering and fuzzy rule interpolation techniques. Inform. Sci., 180: 4772-4783. DOI: 10.1016/j.ins.2010.08.026

Chinchuluun, R., W.S. Lee, J. Bhorania and P.M. Pardalos, 2009. Clustering and Classification Algorithms in Food and Agricultural Applications: A Survey. In: Advances in Modeling Agricultural Systems, Springer, pp: 433-454. DOI: 10.1007/978-0-387-75181-8_21

Christofides, N., M. Mingozzi and P. Toth, 1979. The vehicle routing problem. Combinatorial Optimiz., 11: 315-338.

Das, S., A. Abraham and A. Konar, 2009. Automatic Hard Clustering using Improved Differential Evolution Algorithm. In: Metaheuristic Clustering, Springer, pp: 137-174.

Erol, O.K. and I. Eksin, 2006. A new optimization method: Big bang–big crunch. Adv. Eng. Software, 37: 106-111. DOI: 10.1016/j.advengsoft.2005.04.005

Farag, T.H., W.A. Hassan, H.A. Ayad, A.S. AlBahussain and U.A. Badawi *et al*. 2017. Extended absolute fuzzy connectedness segmentation algorithm utilizing region and boundary-based information. Arabian J. Sci. Eng., 2017: 1-11. DOI: 10.1007/s13369-017-2577-0

Fathian, M., B. Amiri and A. Maroosi, 2007. Application of honey-bee mating optimization algorithm on clustering. Applied Math. Comput., 190: 1502-1513. DOI: 10.1016/j.amc.2007.02.029

Genc, H. and A. Hocaoglu, 2008. Bearing-only target tracking based on big bang-big crunch algorithm. Proceedings of the 3rd International Multi-Conference on Computing in the Global Information Technology, Jul. 27-Aug. 1, IEEE Xplore Press, Athens, Greece, pp: 229-233. DOI: 10.1109/ICCGI.2008.53

Glover, F., M. Laguna and R. Marti, 2002. Scatter Search. In: Theory and Applications of Evolutionary Computation: Recent Trends, Ghosh, A. and S. Tsutsui (Eds.), Springer-Verlag, pp: 519-529.

Gonzalez, T.F., 1982. On the Computational Complexity of Clustering and Related Problems. In: System Modeling and Optimization, Drenick, R.F. and F. Kozin (Eds.), Springer, pp: 174-182.

Greistorfer, P. and S. Voß, 2005. Controlled pool maintenance in combinatorial optimization. Proceedings of the Conference on Adaptive Memory and Evolution: Tabu Search and Scatter Search, University of Mississippi, Kluwer Academic Publishers, pp: 387-424.

Grubesic, T.H., 2006. On the application of fuzzy clustering for crime hot spot detection. J. Quantitative Criminol., 22: 77. DOI: 10.1007/s10940-005-9003-6

Güngör, Z. and A. Ünler, 2007. K-harmonic means data clustering with simulated annealing heuristic. Applied Math. Comput., 184: 199-209. DOI: 10.1016/j.amc.2006.05.166

Halberstadt, W. and T.S. Douglas, 2008. Fuzzy clustering to detect tuberculous meningitis-associated hyperdensity in CT images. Comput. Biol. Med., 38: 165-170. DOI: 10.1016/j.compbiomed.2007.09.002

Hasançebi, O. and S.K. Azad, 2012. An exponential big bang-big crunch algorithm for discrete design optimization of steel frames. Comput. Structures, 110: 167-179. DOI: 10.1016/j.compstruc.2012.07.014

Hatamlou, A., 2013. Black hole: A new heuristic optimization approach for data clustering. Inform. Sci., 222: 175-184. DOI: 10.1016/j.ins.2012.08.023

Hatamlou, A., S. Abdullah and H. Nezamabadi-Pour, 2012. A combined approach for clustering based on K-means and gravitational search algorithms. Swarm Evolut. Comput., 6: 47-52. DOI: 10.1016/j.swevo.2012.02.003

Hatamlou, A., S. Abdullah and M. Hatamlou, 2011. Data Clustering using Big Bang–Big Crunch Algorithm. In: Innovative Computing Technology, Springer, pp: 383-388.

Jain, A.K., 2010. Data clustering: 50 years beyond K-means. Patt. Recognit. Lett., 31: 651-666. DOI: 10.1016/j.patrec.2009.09.011

Jaradat, G.M. and M. Ayob, 2013. Effect of elite pool and Euclidean distance in Big Bang-Big Crunch metaheuristic for post-enrolment course timetabling problems. Int. J. Soft Comput., 8: 96-107. DOI: 10.3923/ijscomp.2013.96.107

Jaradat, G.M., A. Al-Badareen, M. Ayob, M. Al-Smadi and I. Al-Marashdeh *et al*., 2018. Hybrid elitist-ant system for nurse-rostering problem. J. King Saud Univ. Comput. Inform. Sci. DOI: 10.1016/j.jksuci.2018.02.009

■■

Jensi, R. and G.W. Jiji, 2015. Hybrid data clustering approach using k-means and flower pollination algorithm. arXiv preprint arXiv:1505.03236, 2015.

Karaboga, D. and C. Ozturk, 2011. A novel clustering approach: Artificial Bee Colony (ABC) algorithm. Applied Soft Comput., 11: 652-657. DOI: 10.1016/j.asoc.2009.12.025

Kaveh, A. and S. Talatahari, 2009. Size optimization of space trusses using big bang-big crunch algorithm. Comput. Structures, 87: 1129-1140. DOI: 10.1016/j.compstruc.2009.04.011

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

Kuo, R., Y. Syu, Z.Y. Chen and F.C. Tien, 2012. Integration of particle swarm optimization and genetic algorithm for dynamic clustering. Inform. Sci., 195: 124-140. DOI: 10.1016/j.ins.2012.01.021

Li, J., K. Wang and L. Xu, 2009. Chameleon based on clustering feature tree and its application in customer segmentation. Annals Operat. Res., 168: 225-245. DOI: 10.1007/s10479-008-0368-4

Liu, R., L. Jiao, X. Zhang and Y. Li, 2012. Gene transposon based clone selection algorithm for automatic clustering. Inform. Sci., 204: 1-22. DOI: 10.1016/j.ins.2012.03.021

Liu, Y., Z. Yi, H. Wu, M. Ye and K. Chen, 2008. A tabu search approach for the minimum sum-of-squares clustering problem. Inform. Sci., 178: 2680-2704. DOI: 10.1016/j.ins.2008.01.022

Mester, D. and O. Bräysy, 2007. Active-guided evolution strategies for large-scale capacitated vehicle routing problems. Comput. Operat. Res., 34: 2964-2975. DOI: 10.1016/j.cor.2005.11.006

Mitra, S. and P.P. Kundu, 2011. Satellite image segmentation with shadowed C-means. Inform. Sci., 181: 3601-3613. DOI: 10.1016/j.ins.2011.04.027

Park, N.H., S.H. Oh and W.S. Lee, 2010. Anomaly intrusion detection by clustering transactional audit streams in a host computer. Inform. Sci., 180: 2375-2389. DOI: 10.1016/j.ins.2010.03.001

Prayogo, D., M.Y. Cheng, Y.W. Wu, A.A. Herdany and H. Prayogo, 2018. Differential big bang-big crunch algorithm for construction-engineering design optimization. Automat. Construct., 85: 290-304. DOI: 10.1016/j.autcon.2017.10.019

Qin, A. and P. Suganthan, 2004. A robust neural gas algorithm for clustering analysis. Proceedings of the International Conference on Intelligent Sensing and Information Processing, Jan. 4-7, IEEE Xplore Press, Chennai, India, pp: 342-347. DOI: 10.1109/ICISIP.2004.1287680

Resende, M.G., C.C. Ribeiro, F. Glover and R. Martí, 2010. Scatter Search and Path-Relinking: Fundamentals, Advances and Applications. In: Handbook of Metaheuristics, Springer, pp: 87-107.

Rochat, Y. and É.D. Taillard, 1995. Probabilistic diversification and intensification in local search for vehicle routing. J. Heurist., 1: 147-167. DOI: 10.1007/BF02430370

Selim, S.Z. and M.A. Ismail, 1984. K-means-type algorithms: A generalized convergence theorem and characterization of local optimality. IEEE Trans. Patt. Anal. Machine Intell., PAMI-6: 81-87. DOI: 10.1109/TPAMI.1984.4767478

Senthilnath, J., S. Omkar and V. Mani, 2011. Clustering using firefly algorithm: Performance study. Swarm Evolut. Comput., 1: 164-171. DOI: 10.1016/j.swevo.2011.06.003

Song, Y.C., H.D. Meng, M.J. O'Grady and G.M. O'Hare, 2010. The application of cluster analysis in geophysical data interpretation. Comput. Geosci., 14: 263-271. DOI: 10.1007/s10596-009-9150-1

Szeto, W.Y, Y. Wu and S.C. Ho, 2011. An artificial bee colony algorithm for the capacitated vehicle routing problem. Eur. J. Operat. Res., 215: 126-135. DOI: 10.1016/j.ejor.2011.06.006

Talbi, E.G., 2009. Metaheuristics: From Design to Implementation. 1st Edn., John Wiley and Sons, Hoboken, ISBN-10: 0470496908, pp: 500.

Tsai, C.Y. and I.W. Kao, 2011. Particle swarm optimization with selective particle regeneration for data clustering. Expert Syst. Applic., 38: 6565-6576. DOI: 10.1016/j.eswa.2010.11.082

Yeh, W.C. and C.M. Lai, 2015. Accelerated simplified swarm optimization with exploitation search scheme for data clustering. PloS One, 10: e0137246-e0137246. DOI: 10.1371/journal.pone.0137246

Zhang, L. and Q. Cao, 2011. A novel ant-based clustering algorithm using the kernel method. Inform. Sci., 181: 4658-4672. DOI: 10.1016/j.ins.2010.11.005