

Local search algorithms based on benchmark test functions problem

Atheer Bassel, Hussein M. Haglan, Akeel Sh. Mahmoud

Computer Center, University of Anbar, Iraq

Article Info

Article history:

Received Mar 18, 2020

Revised May 14, 2020

Accepted May 26, 2020

Keywords:

Benchmark test functions

Great deluge

Local search algorithm

Meta-heuristic

Optimization

Simulated annealing

ABSTRACT

Optimization process is normally implemented to solve several objectives in the form of single or multi-objectives modes. Some traditional optimization techniques are computationally burdensome which required exhaustive computational times. Thus, many studies have invented new optimization techniques to address the issues. To realize the effectiveness of the proposed techniques, implementation on several benchmark functions is crucial. In solving benchmark test functions, local search algorithms have been rigorously examined and employed to diverse tasks. This paper highlights different algorithms implemented to solve several problems. The capacity of local search algorithms in the resolution of engineering optimization problem including benchmark test functions is reviewed. The use of local search algorithms, mainly Simulated Annealing (SA) and Great Deluge (GD) according to solve different problems is presented. Improvements and hybridization of the local search and global search algorithms are also reviewed in this paper. Consequently, benchmark test functions are proposed to those involved in local search algorithm.

This is an open access article under the CC BY-SA license.



Corresponding Author:

Atheer Bassel,

Computer Center,

University of Anbar, Iraq.

Email: atheerbassel@siswa.ukm.edu.my

1. INTRODUCTION

Artificial Intelligence (AI) encompasses technologies that are expanded from the word 'technology,' and technology is a word originating from 'techné'; a Greek word that carries the meaning of "art" and "skill" [1-3]. Meanwhile, a refined technology encompasses a growing establishment of well-read and well-refined processes and skill. In the past decades, the use of meta-heuristic techniques has been expansive, particularly in the resolution of a lot of real-life applications including economic systems, management, communication, course-timetabling problems, scheduling, marketing strategies, production planning, logistics, design and routing [4-6]. Using these methods, the best solution for a given problem is generated, through the enumeration of all conceivable solutions of the full search space [7-9].

In the last few years, diverse algorithms have been used in diverse domains that relate to expert and intelligent systems, and these include management systems of water resources, benchmark test functions, and modelling the distribution of signal strength in communication systems [10-11]. For the already addressed problems, it appears that their optimal solutions appeared to be known as priori and were presented in the pre-specified early domains. Somehow, it is common that the optimal solutions are not already known for practical problems in expert and intelligent systems. Additionally, due to the intricacy of function of real-life situations, estimating the optimal solution is at times difficult to do. Hence, these algorithms will not

be as useful when the initial domains carry no optimal solution. Having precise estimation of the initial domains beforehand is therefore highly crucial [12-13].

The performance of diverse metaheuristics concerning the post-enrolment course timetabling problems has been empirically addressed in many studies [14-17]. In this regard, a comparison between metaheuristics must be fair, and so, the problem requirements to be addressed become the study's focal point. For this purpose, similar problem-specific knowledge and equal settings of parameter are employed for all metaheuristics [8]. The settings of the parameter were established after series of appropriate parameter settings for each algorithm have been examined. The two criteria of conflict in single-based and population-based meta-heuristic can accordingly be observed in Figure 1. Single-based meta-heuristic typically has more capacities in exploitation while population-based heuristic appears to be more adept in terms of abilities of exploration [18]. The single-based and population-based meta-heuristic as well as their well-known techniques are further elaborated next.

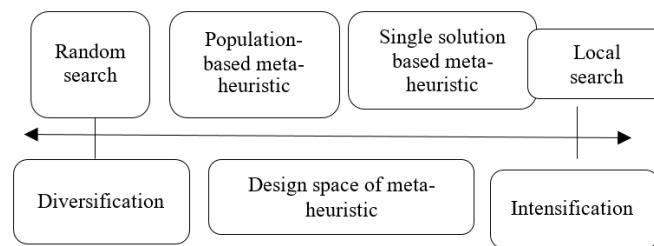


Figure 1. The conflict between diversification and intensification [9]

Improvement of local search algorithms (e.g., TS, SA, GD and HC) has been reported in the literature. Besides that, attempts to improve the algorithms' performance through the enhancement of several of these parts were also demonstrated. Unfortunately, the review of the extant literature demonstrates the presence of information that is vague and inconsistent, particularly the resolution of the problem of benchmark test functions.

This paper presents the local search algorithms based on benchmark test functions problem. Besides that, the improvement on the diverse divisions of local search algorithms for instance GD and SA grounded upon the benchmark test problems is scrutinised in this paper, in order to furnish valuable references and guidance to the relevant scholars. Following the introductory part of this paper, the next topic to be detailed out is the single base solution meta-heuristics. Then, in section 3, experiments executed on some selected benchmarks test function are illustrated. The final section concludes the paper.

2. SINGAL-BASE SOLUTION META HEURISTIC

Single-based meta-heuristic (SBH) or also dubbed as local search, encompasses a procedure which begins with a preliminary solution. At each iteration, a neighborhood operator is employed in the generation of an adjacent solution [19-20]. Single-based metaheuristics denote the methods of search which discompose one solution at a given time [21-22]. Examples of single-based methods include Tabu Search (TS), Simulated Annealing (SA), and Search of Variable Neighborhood (SVN). The single point-based methods usually begin at certain initial solution. Here, during each iteration, the present solution is equated and substituted by a neighbor, leading to the improvement of the objective function. Defined Neighborhood Structure is employed for the purpose. Then, when all candidate neighbors are inferior to the existing solution, the searching ends, because at this stage, a local optimum is already achieved [2]. Several metaheuristics with the capacity of manipulating just one solution at one time are discussed in the next subsections. These local search routines have the capacity in manipulating the solution space toward solutions of better quality. They are also able to intensify the search by efficiently searching for solutions of good quality. Accordingly, the extant literature shows the applications of these algorithms in the resolution of diverse problems associated with optimization and benchmark test functions problem.

3. GREAT DELUGE

The algorithms of great deluge (GD) by Duek [10] are comparable to those of SA except that the primary purpose of GD algorithms is to substitute those of SA. The parameters used in GD algorithms are

less in amount, compared to those employed in SA algorithms, but like SA algorithms, GD algorithms also always accept improved solutions while the worsening ones are accepted according to certain likelihood. GD algorithms accept worsening solutions in order to escape basin attraction (local optima) [23]. The applications of GD algorithms can be seen in many problems of optimization for instance those related to education timetabling such as timetabling of examinations, courses and school [24], problems related to sport timetabling and those related to Transportation timetabling. Burke et al. [25] indicated that only two parameters are needed when using GD algorithms as follows: time of search which is denoted by number of iterations, and level of quality of estimated solution. The entire procedures are then repeated for a given number of iteration as shown in the following algorithm 1.

Algorithm 1: Pseudo_code of the Basic GD algorithm

```

Determine initial candidate solution  $S$ ;
Determine initial level  $L$ ;
 $S = S_0$ ; // Generation of the initial solution
Choose the rain speed decay _rate; // decay_rate > 0
Choose the initial water level  $Level$ ;
Repeat
    Generate a random neighbor solution  $S'$ 
    If  $f(S') < Level$ :
         $S = S'$  // accept the neighbor solution;
         $Level = level - decay\_rate$ ; // update the water
level until termination criterion is satisfied
Until stopping criteria
Output: Best solution found.
```

The parameter (i.e. level) fully determines the quality of the attained outcomes and search time. In this regard, high rain speed parameter (decay rate) value will result in speedy algorithm but the obtained outcomes will have poor quality, while using the small rain speeds (decay rate) value will result in more superior algorithm which generate superior results within a greater computational time.

4. SIMULATED ANNEALING (SA)

The algorithms of simulated annealing (SA) by Kirkpatrick et al. [26] simulates the method of hill climbing except that it also carries a mechanism for escaping the local optima. In illustrating this algorithm, annealing entails the physical process of heating up a solid using high temperature and gradually cooling it down to a point where the solid crystallizes, with no more occurrence of changes while the system accomplishes a stable state. In other words, thermalisation has been achieved. Accordingly, a sequence of temperatures employed in the system thermalisation entails an annealing schedule or cooling schedule. The algorithm of SA was therefore based on the annealing process, and this algorithm is employed in addressing the problems associated with combinatorial optimization. SA begins from an initial solution that is produced randomly. During the process, the randomly chosen neighbour (move) of the existing solution is assessed, and the improved moves concerning the objective function are generally accepted while the worse moves are accepted but with some probability ascertained using the Boltzmann probability P which is computed using formula of $P = e^{-\Delta E / T}$, with ΔE representing the difference of the evaluation of objective function between the existing and the candidate solutions and T representing a parameter (called the temperature) that declines sporadically during the process of search following certain cooling schedule. In general, the cooling schedule choice affects the final solution particularly in terms of its quality. In this regard, cooling schedule that is faster will result in local optima that is faster as well [27]. Meanwhile, the use of slower cooling schedule will require a search that is more comprehensive, resulting in solution of higher quality. However, the use of slower cooling will consume more time. Many methods can be employed in the reduction of temperature during the search process. For instance, a progress formula (geometric cooling schedule) can be employed. Similarly, the temperature can be reduced using certain number of moves or just the successful moves.

The effective application of SA algorithm has been observed in problems optimization for instance, problems of education timetabling such as timetabling of courses, examinations, and school, and problems of sport and transportation timetabling [28]. The subsequent pseudo-code is used in the establishment of the SA heuristic. As can be observed above, the implementation begins with state s_0 and it ensues until the maximum of k_{max} steps are reached or until a state comprising energy e_{max} or less is discovered. In this situation, the call neighbour(s) should produce a randomly selected neighbour of certain state s , and a random value should be returned by the call random and the value should fall in the range $[0, 1]$. The call temp (r) defines the annealing schedule and this should generate the temperature to be used correspondingly to the fraction r of the expanded time budget. The operation of SA algorithm is commenced

by the presence of initial solution x , and then, the solution is improved through the production of a candidate solution. The candidate solution would be accepted if it is superior to solution x , or, it will be accepted using the acceptance criterion that relies on a certain probability r (r encompasses a random number between 0 and 1). If r is lower than the value of (where entails the difference between the candidate and the existing solutions, while T denotes a control parameter known as temperature), the candidate solution is then accepted. Also, following a cooling schedule (β encompasses the cooling rate), the temperature is progressively reduced. This process recurs and stops after the criterion of termination is achieved. Accordingly, the SA pseudo-code is shown below.

Algorithm 2: Pseduo_code of Basic SA algorithm

```

Generate the initial solution  $X = X_0$ 
Set the starting temperature  $T = T_0$ 
Set Cooling rate  $\beta$ 
Set max number of trials each temperature  $i$ ;
Repeat
     $i = 0$ 
    For  $i = 0$  to  $i_{max}$  do
        Generate a random neighbor
         $\Delta E = f(X') - f(X)$ 
        If  $\Delta E \leq 0$ 
             $X = X'$  // accept the neighbor solution
        Else
            Generate a random number between 0 and 1 ( $r$ )
            If ( $r < e^{-\Delta E/T}$ )
                 $X = X'$ 
    end
    end  $i = i - 1$ 
    end
    Reduce temperature rate  $T = T * \beta$ 
    Update the best solution
Until stopping criteria is satisfied  $T = 0$ 
Return Best found solution.
  
```

5. EXPERIMENT

The present section presents the execution of a group of experiments on several benchmarks obtained from [28]. The experiments were to determine if the local search algorithms were indeed implemented in the studies. This was followed by the assessment and comparison of the proposed GD and SA algorithms, using an ensemble of expansively employed unimodal and multimodal standard benchmark functions. Table 1 tabulates 8 standard benchmark functions that have been expansively employed in past works particularly those that attempted to test the general-purpose optimization algorithms in terms of performance-wise. Here, the test functions in this group alongside their characteristics and dimension are displayed. Included in these benchmark functions are functions with different properties for instance, unimodal and multimodal. Table 2 tabulates a comparison between SA and GD.

Table 1. Benchmark test functions (*continue*)

Property	Peak function	Formula	Search space
Uni-modal	Cone	$f(x) = \sqrt{\sum_{i=1}^n x_i^2}$	[-100,100]
Uni-modal	Sphere	$f(x) = \sum_{i=1}^n x_i^2$	[-5.12, 5.12]
Uni-modal	Schwefel 2.22	$f(x) = \sum_{i=1}^n x_i + \prod_{i=1}^n x_i $	[-10, 10]
Uni-modal	Quadric	$f(x) = \sum_{i=1}^n (\sum_{j=1}^i x_j)^2$	[-1.28, 1.28]
Multi-modal	Rastrigin	$f(x) = \sum_{i=1}^n (x_i^2 - 10 \cos 2\pi x_i + 10)$	[-5.12, 5.12]
Multi-modal	Griewank	$f(x) = \frac{1}{4,000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	[-600, 600]

Table 1. Benchmark test functions

Property	Peak function	Formula	Search space
Multi-modal	Ackley	$f(x) = -20 \exp(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}) - \exp(\frac{1}{n} \sum_{i=1}^n \cos 2\pi x_i) + 20$	[-32, 32]
Multi-modal	Weierstrass	$f(x) = \sum_{i=1}^n (\sum_{k=0}^{k_{max}} [a^k \cos(2\pi b^k (xi + 0.5))]) - n \sum_{k=0}^{k_{max}} [a^k \cos(\pi b^k)]$	[-0.5, 0.5]

Table 2. Comparison of SA and GD

Algorithm	Performance	Processing time	Parameters
Great Deluge	Suitable in search space exploration because it takes in the worse solutions following non-improvement of several iterations.	Has better speed than Simulated Annealing	<ul style="list-style-type: none"> - No. of iterations - Solution quality estimation
Simulated Annealing	Suitable in search space exploration because it takes in the worse solutions following non-improvement of several iterations. based on probabilistic conditions	Speed is less than that of Great Deluge	<ul style="list-style-type: none"> - No. of iterations - Initial temperature - Final temperature - Decreasing rate

6. CONCLUSION

The algorithms of Great Deluge and Simulated Annealing have been presented in this paper with their performance are examined. As can be observed, both techniques have robustness and capacity in resolving countless of actual life problems. Great Deluge has better speed compared to simulated annealing because Great Deluge has less used parameters (two parameters) as opposed to simulated annealing that employs four parameters. Somehow, recent studies suggested that hybridizing two or more algorithms can increase the capacity of the resultant algorithm in solving actual life problems. As future scrutiny, these algorithms can also be recommended in the resolution of the benchmark test function.

ACKNOWLEDGEMENTS

This research has been made possible through research grants DIP-2016-018 awarded by Universiti Kebangsaan Malaysia (UKM) and Ministry of Higher Education, Malaysia. The researchers wish to express their gratitude to these institutions for their support.

REFERENCES

- [1] Z. Boulouard, A. El Haddadi, F. Bouhafer, A. El Haddadi, L. Koutti, and B. Dousset, "Bat-cluster: A bat algorithm-based automated graph clustering approach," *Int. J. Electr. Comput. Eng.*, vol. 8, no. 2, pp. 1122-1130, 2018.
- [2] A. K. Ariyani, W. F. Mahmudy, and Y. P. Anggodo, "Hybrid Genetic Algorithms and Simulated Annealing for Multi-trip Vehicle Routing Problem with Time Windows," *Int. J. Electr. Comput. Eng.*, vol. 8, no. 6, p. 4713, 2018.
- [3] B. Freisleben and P. Merz, "Genetic local search algorithm for solving symmetric and asymmetric traveling salesman problems," *Proc. IEEE Conf. Evol. Comput.*, pp. 616-621, 1996.
- [4] Z. Friggstad, M. Rezapour, and M. R. Salavatipour, "Local search yields a PTAS for k-means in doubling metrics," *SIAM J. Comput.*, vol. 48, no. 2, pp. 452-480, 2019.
- [5] W. N. A. D. Abed, O. A. Imran, and I. S. Fatah, "Automatic generation control based whale optimization algorithm," *Int. J. Electr. Comput. Eng.*, vol. 9, no. 6, pp. 4516-4523, 2019.
- [6] W. R. Abdul-Adheem, "An enhanced particle swarm optimization algorithm," *Int. J. Electr. Comput. Eng.*, vol. 9, no. 6, pp. 4904-4907, 2019.
- [7] G. R. Raidl, J. Puchinger, and C. Blum, *Metaheuristic Hybrids. Handbook of Metaheuristics* 2010.
- [8] B. Madadi, R. van Nes, M. Snelder, and B. van Arem, "A bi-level model to optimize road networks for a mixture of manual and automated driving: An evolutionary local search algorithm," *Comput. Civ. Infrastruct. Eng.*, vol. 35, no. 1, pp. 80-96, 2020.
- [9] T. Albash and D. A. Lidar, "Demonstration of a Scaling Advantage for a Quantum Annealer over Simulated Annealing," *Phys. Rev. X*, vol. 8, no. 3, p. 31016, 2018.
- [10] S. K. Joshi and J. C. Bansal, "Parameter tuning for meta-heuristics," *Knowledge-Based Syst.*, vol. 189, no. xxxx, p. 105094, 2020.

- [11] D. Kizilay, M. F. Tasgetiren, Q. Pan, and G. Sürer, "ScienceDirect ScienceDirect An Ensemble Ensemble of of Meta-Heuristics Meta-Heuristics for the the Energy-Efficient Energy-Efficient Blocking Blocking Flowshop Flowshop Scheduling Scheduling Problem Problem," *Procedia Manuf.*, vol. 39, no. 2019, pp. 1177-1184, 2020.
- [12] F. Schäfer, J.-H. Menke, and M. Braun, "Comparison of Meta-Heuristics for the Planning of Meshed Power Systems," 2020.
- [13] P. Singh, M. Dutta, and N. Aggarwal, "A review of task scheduling based on meta-heuristics approach in cloud computing," *Knowl. Inf. Syst.*, vol. 52, no. 1, 2017.
- [14] C. Blum and A. Roli, "Metaheuristics in combinatorial optimization: Overview and conceptual comparison," *ACM Comput. Surv.*, vol. 35, no. 3, pp. 268–308, 2003.
- [15] E. Talbi, Metaheuristics: from design to implementation. *Knowledge and Information Systems McGill Journal of Education* 2009.
- [16] M. Dorigo, "No Optimization, Learning and Natural Algorithms", Earning and Natural Algorithms, *International Journal of Intelligence Science*, Vol.4 No.41992.
- [17] Yang XS. (2013) Metaheuristic Optimization: Nature-Inspired Algorithms and Applications. In: Yang XS. (eds) Artificial Intelligence, Evolutionary Computing and Metaheuristics. Studies in Computational Intelligence, vol 427. Springer, Berlin, Heidelberg.
- [18] Z. Michalewicz and D. B. Fogel, How to solve it: modern heuristics. *Springer Science & Business Media* 2013.
- [19] Z. Yu et al., "Adaptive Multimodal Continuous Ant Colony Optimization," *IEEE Trans. Evol. Comput.*, vol. 21, no. 2, pp. 191-205, 2017.
- [20] S. Alani, Z. Zakaria, and H. Lago, "A new energy consumption technique for mobile Ad-Hoc networks," *Int. J. Electr. Comput. Eng.*, vol. 9, no. 5, pp. 4147-4153, 2019.
- [21] C. P. S. A. János Fodor, Ryszard Klempous, Recent Advances in Intelligent Engineering Systems. (Vol. 378). *Springer Science & Business Media*. 2011.
- [22] G. Zäpfel, R. Braune, and M. Bögl, Metaheuristic search concepts: A tutorial with applications to production and logistics. *Springer Science & Business Media*. 2010.
- [23] M. Chiarandini, K. Socha, M. Birattari, and O. R. Doria, International Timetabling Competition A Hybrid Approach. *International Timetabling Competition* 2003.
- [24] G. DUECK, "New optimization heuristics: The great deluge algorithm and the record-to-record travel," *J. Comput. Phys.*, vol. 104, no. 1, pp. 86-92, 1993.
- [25] M. Oktavia, A. Aman, and T. Bakhtiar, "Courses timetabling problem by minimizing the number of less preferable time slots," in IOP Conf. Series: Materials Science and Engineering, 2017, vol. 166, no. 1, p. 12025.
- [26] S. Kirkpatrick, C. D. G. Jr., and M. P. Vecchi, "Optimization by simulated annealing," *Science* (80-.), vol. 220, no. 4598, pp. 671-680, 1983.
- [27] R. A. Oude Vrielink, E. A. Jansen, E. W. Hans, and J. van Hillegersberg, "Practices in timetabling in higher education institutions: a systematic review," *Springer US*, 2017.
- [28] Bassel, A., & Nordin, M. J., Mutation and memory mechanism for improving Glowworm Swarm Optimization algorithm. In *2017 IEEE 7th Annual Computing and Communication Workshop and Conference (CCWC)*, pp.17.

BIOGRAPHIES OF AUTHORS



Atheer bassel received the B.Sc. degree in computer science from University of Anbar, Iraq, in 2004, the M.Sc. degree in computer science, in 2011 and the Ph. D degree in computer science from Universiti Kabangsaan Malaysia (UKM) in 2018. He is currently an assistant professor in the Department of Computer Science, Faculty of Computer Science and Information Technology, University of Anbar, Iraq. His main research area are metaheuristics, optimization algorithms, Image processing and their applications.



Akeel Shaker was born in Iraq, in 1981. Received the B.Sc. degree in computer science from university of anbar, Iraq, in 2003, the M.Sc. degree in computer science, in 2017 from Belgorod University in the Russia. He is currently an assistant Teacher in the Department of Computer Science, Faculty of Computer Science and Information Technology, university of anbar, Iraq. His main research area is wireless communication, computer networks, and networking Technology.



Hussein Mahdi Haglan was born in Al-Anbar Province, Iraq in 1984. He received the B.Sc. degree in Information System from the University of Anbar, Ramadi, Iraq in 2006 and the master's degree in information technology from University Tenaga Nasional (UNITEN), Malaysia in 2014. He is currently one of the teaching staff at the University of Anbar, Iraq. His research interests include data communication and networks, big data, database, and artificial intelligence.