Open Access
Baghdad Science Journal Vol.16 (Special Issue) 2019

DOI: http://dx.doi.org/10.21123/bsj.2019.16.2(SI).0445

Taxonomy of Memory Usage in Swarm Intelligence-Based Metaheuristics

Shaymah Akram Yasear[*] *Ku Ruhana Ku-Mahamud*[**]

Received 5/9/2018, Accepted 31/10/2018, Published 20/6/2019

Abstract:

Metaheuristics under the swarm intelligence (SI) class have proven to be efficient and have become popular methods for solving different optimization problems. Based on the usage of memory, metaheuristics can be classified into algorithms with memory and without memory (memory-less). The absence of memory in some metaheuristics will lead to the loss of the information gained in previous iterations. The metaheuristics tend to divert from promising areas of solutions search spaces which will lead to non-optimal solutions. This paper aims to review memory usage and its effect on the performance of the main SI-based metaheuristics. Investigation has been performed on SI metaheuristics, memory usage and memory-less metaheuristics, memory characteristics and memory in SI-based metaheuristics. The latest information and references have been further analyzed to extract key information and mapped into respective subsections. A total of 50 references related to memory usage studies from 2003 to 2018 have been investigated and show that the usage of memory is extremely necessary to increase effectiveness of metaheuristics by taking the advantages from their previous successful experiences. Therefore, in advanced metaheuristics, memory is considered as one of the fundamental elements of an efficient metaheuristic. Issues in memory usage have also been highlighted. The results of this review are beneficial to the researchers in developing efficient metaheuristics, by taking into consideration the usage of memory.

Keywords: Global optimization, Memory usage, Nature-inspired metaheuristic, Optimization algorithm, Search experience.

Introduction:

Optimization problems can be solved using two methods, namely, exact and approximate methods. Exact methods ensure optimal solutions but the run-times often increase dramatically with problem complexity. In contrast, approximate methods can be used to find feasible solutions in a reasonable time. However, there is no guarantee that the optimal solution will be found. Approximate methods can be divided into two sub-categories, namely, heuristic algorithms and metaheuristics. In general, heuristics are very specific to the problems they aim to solve. Therefore, the use of more flexible heuristics is required. Metaheuristics are more generic methods in solving various optimization problems. Metaheuristics have received increasing interest and have shown their effectiveness in broad areas of application by solving many optimization problems.

School of Computing, Universiti Utara Malaysia, Malaysia
Corresponding author:
[*]shayma.akram.yasear@gmail.com
[**]ruhana@uum.edu.my

Metaheuristics have several sources of inspiration; the main source is nature. Most natureinspired metaheuristics mimic biological or physical phenomena (1). Based on the source of inspiration, metaheuristics can be divided into biological- and physical-based metaheuristics. Under each category, the metaheuristics can be divided into evolutionary algorithms (EAs) and swarm intelligence (SI) algorithms (2). EAs correspond to a group of metaheuristics that inspired by the natural selection theory proposed by Darwin (3). EAs apply the principles of survival and evolution of the strongest individuals to produce better approximations to a solution. Some algorithms that belong to this category are genetic algorithm (GA) (4), differential evolution (DE) (5), evolutionary strategy (ES) (6) and biogeography-based optimization (BBO) algorithm (7). Swarm intelligence was first introduced by Beni and Wang (8), the algorithms belong to SI are based on the theory of collective behavior in self-organized systems and considered as effective methods for finding approximate solutions to complex optimization problems. Therefore, they have earned more popularity compared to other population and EA methods (9).

The main components in any metaheuristic are the diversification and intensification strategies (2). Diversification directs the research process to areas that have not yet been explored, with the aim of detecting new and better solutions that are different from those previously encountered. Intensification strategy aims to find the best solution in promising regions. The balance between these strategies is essential for a metaheuristic to provide high quality solutions. Historical information collected by the algorithm during the search process can be used to control the diversification and intensification strategies. This information is considered as a memory which is used by a metaheuristic to carry out the search process (2).

This paper aims to highlight the usage of memory in the main metaheuristics that belong to the SI class, namely, ant colony optimization (ACO) (10), particle swarm optimization (PSO) (11), artificial bee colony (ABC) (12), Cat swarm optimization (CSO) (13) firefly algorithm (FA) (14), bat algorithm (BA) (15) and Grey wolf optimizer (GWO) (16). This includes the discussions on the content of memory and its impact on the performance of an algorithm. Output of this review can be used in developing new metaheuristics with respect to the usage of memory. The next section of the paper presents a brief introduction to SI-based metaheuristics. The third section discusses the main differences between memory and memory-less metaheuristics. Discussions on the memory characteristics and memory in SI-based metaheuristics are provided in the fourth and fifth sections. Concluding remarks are presented in the final section.

Swarm Intelligence-Based Metaheuristics:

Most SI-based metaheuristics are imitations of the behavior ants, termites, bees, fish and birds (17). Some SI-based metaheuristics are inspired by the same creature with enhancement made for better performance (18-23). However, the major difference between these metaheuristics is in the moving rules of individuals in the solutions space. Not all SI-based metaheuristics are based on biological systems. There are other SI-based metaheuristics which are inspired by physical and chemical systems, such as the gravitational search algorithm (24). All swarm intelligence metaheuristics are population-based and composed of simple agents interacting with each other and the environment following simple rules, which lead to an intelligence global behavior (25). A number of SI-based metaheuristics have been proposed and they have shown superior skills in solving various optimization problems (12, 14-16, 26, 27). Figure 1 depicts the chronology of SI-based metaheuristics that have been proposed since 1995 until 2017.

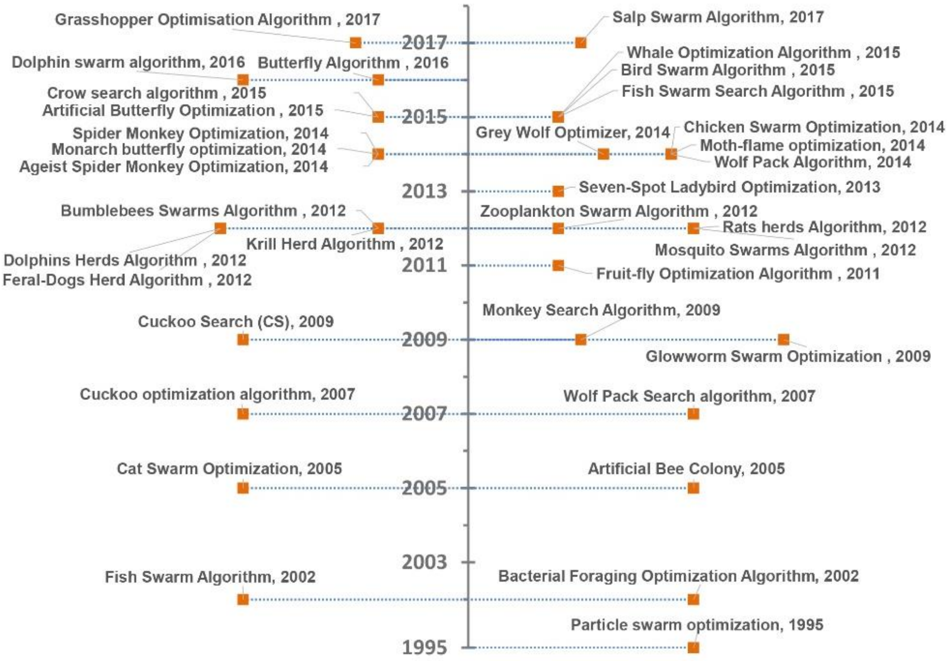

Figure 1. Chronology of swarm-based metaheuristics.

Most nature-inspired metaheuristics were proposed by mimicking biological phenomena. SI-based metaheuristics inspired by biological systems are of great interest in the development of new metaheuristics starting from the year 2013. Their efficiencies have been shown in solving optimization problems in science, engineering and industry fields (28-32). However, not all can produce optimal solution. The free lunch theorem by (33) stated that there is no specific metaheuristic that can solve all types of optimization problems.

Memory Usage Against Memory-Less Metaheuristics:

Metaheuristics can be classified based on the usage of memory into two classes, namely, metaheuristics with memory and memory-less. Memory-less metaheuristics do not use information extracted during the search process. Instead, they execute a Markov process, since the information they need is only the current state of the search process (34). In this case, they tend to guide the search agents outside the promising region of the solution search space. On the other hand, a memory in metaheuristics allows them, while exploring new regions in the search space, to store historical information of the search process. In this case, the algorithm will be able to obtain high quality solutions.

The usage of memory allows a metaheuristic to be effective during both the diversification of the search space and the intensification of promising areas (35). In the diversification process the algorithm will explore new promising regions. The usage of memory will prevent the search process from returning to solution spaces previously explored. On the other hand, intensification strategies use the high quality solutions stored in the memory to focus more on promising regions (36). This helps in reducing the computational cost and gives more robustness to the algorithm. Therefore, memory is considered as one of the basic elements of a great metaheuristic (2).

Memory Characteristics:

The first use of memory (history of the search) in metaheuristics was pioneered by Glover (37). The type of memories can be divided into short-term medium-term and long-term memory classes (38). Short-term memory (STM) also called taboo list, stores most recent candidate solutions generated by the algorithm. The history of the last movements made (points) is stored in a first-in–first-out list. This will prohibit the movement from repeating in the next iteration. In other words, the movement will not be used as long as it remains inside the memory, thus avoiding cycles and trapping in local optima. In metaheuristics, the intensification and diversification strategies can be considered as medium and long term memory respectively (2, 38, 39). The medium-term memory (MTM) stores the information of the best found solutions (elite solutions). This includes optimal or near-optimal potential solutions. This historical information is then used to intensify the search on the regions of the search space with known good fitness function values (35). Long-term memory (LTM) stores information about the regions that have been explored by the algorithm, thus diversifying the search (38, 40, 41).

Based on the type of stored information, the memory can be classified into three main classes, namely, frequency-based, influence-based and quality-based memory. The frequency-based memory stores attributes of the solutions. Based on this it can be divided into transition frequencies, which records the changing attributes during the search process and residence frequencies, which records how often the attributes can characterize the produced solutions. The influence-based memory stores information related to the effects of the decisions made in the quality and structure of the solution. Quality-based memory records shared attributes such as promising paths in good solutions. This helps the search process in moving away from poor solutions (Fig.2).

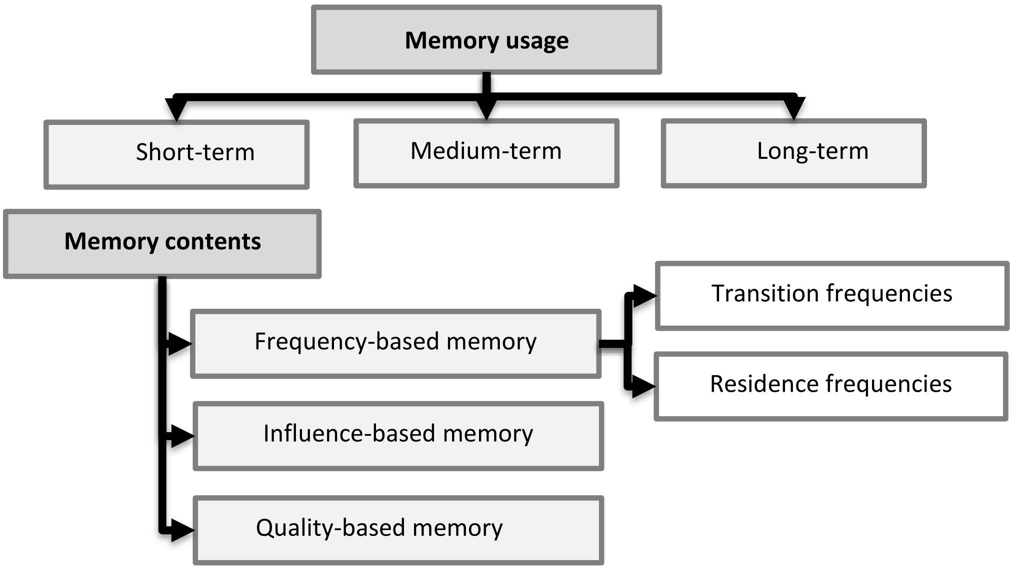

Figure 2. Taxonomy of Memory based on Usage and Stored Information.

The memory stores the information based on the learning process and a learning process uses a memory to record the information (42). This helps a metaheuristic in solving complex optimization problems. Integrating the learning process into a metaheuristic and using historical information stored in the memory during the search process is known as reactive search. This represents the ability of the algorithm to be adapted based on different situations with respect to feedback based on its past experience during the search process. This can be considered as an online adaptation (43).

Memory in Swarm-Based Metaheuristics:

One important SI algorithm is the ant colony optimization (ACO) algorithm which is inspired by the collective foraging behavior of ants (10). ACO uses artificial ants, which mimic the ability of real ants to build invisible paths marked by a chemical produced by ants called pheromone. These trails with traces of pheromone are used by the ants to guide themselves between their nest and a food source. The process in which the real ants mark their paths with traces of pheromone to optimize, collectively, their strategy to search for food is known as *stigmergy* and consists of an indirect form of communication making changes to the environment (44). In ACO the pheromone traces are represented by numerical information that ants use to probabilistically construct solutions for an optimization problem. The pheromone matrix represents a long-term memory, where it is used to store the previously used trails. This allows the ants to perform the *stigmergy* process in the algorithm (18). Furthermore, it provides a short-term memory where it records the short-term attraction to recent rewarding trails (45).

Particle swarm optimization algorithm is a popular SI-based metaheuristic that consists of memory. It was first proposed by Kennedy and Eberhar (11) and originally inspired by the social behaviour of schools of fish and flocks of birds. Each individual (particle) represents a potential solution to an optimization problem. The PSO algorithm seeks to find the optimal solution by moving a set of particles (swarm) in the *n*-dimensional search space. Each particle is represented by a position and a velocity vectors, which determines the direction and distance of movement (flying). In this movement, the particles use historical information (the result of their past exploration), as well as information about their neighbors. This is represented by a memory containing the vector with personal best and global best position found so far in the search space.

Cat swarm optimization proposed by Chu, Tsai (13), was inspired by the hunting behavior of cat, in nature, which includes the tracking and seeking modes. The CSO algorithm will switch between these modes, based on the value of mixture ratio in finding the optimal solution. Cats move based on i) seeking range of the selected dimension (SRD), ii) counts of dimension to change (CDC), iii) self-position considering (SPC) and iv) seeking memory pool (SMP). SRD determines the range of selected dimension to be used for the mutation operation. CDC declares the count of dimensions that will be changed. SPC decides whether the cat will remain in the current position or will be moved to a new one. SMP defines the size of the seeking memory pool, where it determines the positions explored by each cat. The CSO algorithm, also, uses an MTM to store the position of the best cat, which will be used to update the cats' positions (13).

The artificial bee colony (ABC) (12) imitates the foraging behaviour of bees. The colony consists of employed, onlooker, and scout bees and the food sources represent the candidate solutions. The employed bees share information with onlooker bees which wait in the nest and establish a food source (12). The employed bees search for new food sources with respect to the old one. They store the new food-source location in their memory and remove the old one. The scout bees explore the environment surrounding the nest to find new food sources (12). Then, a greedy selection method is applied between the old and candidate solutions to select the better one. The onlooker bees choose food sources by comparing the probability, which is computed based on the fitness value. If a solution does not improve its quality for a certain number of cycles (*limit*), it will be abandoned and replaced by a new random solution, which is discovered by the scout bee.

The light-emitting behaviour of fireflies (46) has inspired (14) to propose the firefly algorithm (FA). In this algorithm, the light intensity (brightness) and attractiveness of a firefly are the key factors that determine its movement. The firefly with higher brightness attracts fireflies with lower brightness to move toward it. The degree of attraction determines the direction and distance of the fireflies. The firefly position is updated based on its attractiveness, controlled by the brightness level. However, the FA is a memory-less algorithm (47). Therefore, the information is not conveyed from one iteration to other iteration. This means that the firefly will not be able to attract other fireflies in successive iterations, because the position of this firefly will also be changed and its information lost (48). Modified versions of the FA were proposed by integrating a memory that records the fireflies with the best solution to be used in the next generation (49) and best solution found so far (47).

The hunting behavior of micro bats, using their echolocation ability, has been used to develop the bat algorithm (BA) by Yang (15). Each flying bat is considered as a potential solution to an optimization problem. The solutions are updated by adjusting its position, velocity and frequency, according to the best position it has reached and what is found by the whole population. At the beginning, the bat has only a small pulse emission rate and a large loudness (15). As the iteration increases, the pulse emission rate increases and the loudness decreases. The local search is controlled by adjusting the rate of pulse emission and loudness based on the proximity of the prey. Thus, the loudness and pulse emission rate are updated if new solutions are enhanced This will direct the bats towards the optimal solution (15). The BA is a memory-less algorithm, as it does not store the best solution found during the optimization process. In this case, the bats tend to escape from the promising regions of the search space. To overcome this limitation, Kiełkowicz and Grela (50) propose a bat algorithm with medium-term memory. The memory is used to store the global best solution found during the optimization process.

Grey wolf optimizer (GWO) by (16) is based on the hunting behaviour of the grey wolf. The hierarchical structure of the grey wolves consists of three leaders, namely, alpha (α), beta (β) and delta (δ). The leaders represent first, second and third best solutions in the search space. The remaining candidate solutions are omega (ω). Leaders' positions are stored in a memory to be used to update the positions of wolves in the n-dimensional space. Searching for prey is an exploration or global search, while attacking prey is exploitation or local search(Table 1).

Table 1. Memory Characteristics of SI-based Metaheuristics.

Algorithm	Memory	Type of memory ST	MT	LT	Memory content
PSO (11)	✓	-	✓	-	Vectors with personal best and global best particles.
ACO (10)	✓	✓	-	✓	Pheromone matrix.
CSO (13)	✓	-	✓	-	The position of the best cat.
ABC (12)	✓	✓	-	✓	Best food source found so far. Food source historical information.
FA (14)	-	-	-	-	-
FA with memory (47, 49)	✓	-	✓	-	Fireflies with best fitness value. Best solution found so far.
BA (15)	-	-	-	-	-
BA with memory (50)	✓	-	✓	-	Best position among all bats.
GWO (16)	✓	-	✓	-	First three best solutions obtained so far.

ST: short-term; **MT**: medium-term; **LT**: long-term

Review of these SI-based metaheuristics shows that most of the SI-based metaheuristics use a MTM. Furthermore, most of these memories store information about the best solution found so-far during the optimization, with respect to the representation of candidate solutions in each algorithm. The ACO and ABC algorithms both used STM and LTM. The pheromone matrix in ACO represents an LTM, while the LTM memory in the ABC stores information about the food source position. On the other hand, the FA and BA are both memory-less algorithms, where they do not use historical information during the search process. However, modified versions of these algorithms were proposed by adding a memory to record the best solutions found so-far. Furthermore, incorporating a memory with memory-less algorithms has shown an improvement in the performance of the FA (49) and BA (50). This implies that the usage of memory is an important role in governing the performance of metaheuristics.

In general, the use of historical information is a crucial point in preventing an algorithm from becoming trapped in local optima and accelerating convergence towards the best solutions. This information is even more important for diversification to avoid becoming trapped in local optima. The usage of MTM can be sufficient to ensure quality of solutions. However, to obtain additional gains in search performance, STM and LTM memory can also be utilized. This helps in maintaining the diversity of population and moves the search toward more promising regions in the search space. Furthermore, this will increase the possibility of finding a global optimal solution for an optimization problem and avoid trapping in local optima. At the same time, it decreases the computational cost by reducing the random exploration of the search space. Furthermore, metaheuristics performance can be further improved by integrating efficient techniques, in terms of the method of saving and retrieving information stored in the memory.

Conclusion:

Metaheuristics can be considered as effective approximate search methods for complex optimization problems in various fields and have an advantage over the exact methods with their ability to produce a feasible solution in a reasonable time. The biology-based metaheuristics are efficient because of their significant capacity to imitate the best characteristics of creatures in nature. Swarm intelligence is one of the most popular categories of metaheuristics and has gained great interest in the research community, due to its applicability in several applications domains. In advanced metaheuristics there is a kind of memory which transfers information from one iteration to the other. However, most do not explicitly use memory, except in the selection of the best solution. The diversification and intensification strategies are the main component in any metaheuristics. The balance between these strategies allows a metaheuristic to discover new promising areas in the search space for a feasible solution to the optimization problem. These strategies can be considered as medium- and long-term memories. The usage of memory is important to ensure that the algorithm will not spend too much time in regions which are already explored. Furthermore, the historical information stored in the memory plays a significant role in improving the performance of metaheuristics by directing the search agents toward the promising region in the search space.

The main issue of memory usage is related to the learning of an algorithm. Normally, it has a dynamic character. In other words, the information contained in the memory should be updated whenever possible. For this reason, it is necessary to consider only relevant information in the learning stage that has to be stored. Sometimes it is convenient to consider only recent information which can be represented by a short-term memory. In other cases, storing the information and its use in the next stage or keeping information related to already checked regions is important. This can be considered as medium- and long-term memories, respectively. Another issue is the determination of attributes of solutions and how long the attributes should be kept in the memory. This is important in developing a metaheuristic that incorporates memory. The presented review results can be of help to other researchers in developing an efficient metaheuristic, by taking into consideration the usage of memory.

Acknowledgment:

The Higher Education Ministry of Malaysia has funded this work under the Fundamental Research Grant Scheme, S/O code 13794.

Conflicts of Interest: None.

References:

1. Siddique N, Adeli H. Nature inspired computing: an overview and some future directions. Cognitive computation Cognit Comput

. 2015;7(6):706-14.

2. Blum C, Roli A. Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM* Computing Surveys ACM Comput Surv. 2003;35(3):268-308.
3. Darwin C. On the origin of species. London: John Murray; 1859.
4. Amjad MK, Butt SI, Kousar R, Ahmad R, Agha MH, Faping Z, *et al*. Recent research trends in genetic algorithm based flexible job shop scheduling problems. Mathematical Problems in Engineering MATH PROBL ENG. 2018;2018.
5. Wu G, Shen X, Li H, Chen H, Lin A, Suganthan PN. Ensemble of differential evolution variants. Information Sciences INFORM SCIENCES. 2018;423:172-86.
6. Bozorg-Haddad O. Advanced optimization by nature-inspired algorithms: Springer; 2018.
7. Guo W, Chen M, Wang L, Mao Y, Wu Q. A survey of biogeography-based optimization. Neural Computing and Applications Neural Comput Appl. 2017;28(8):1909-26.
8. Beni G, Wang J. Swarm intelligence in cellular robotic systems. In: Dario P, Sandini G, Aebischer P. editors. Robots and Biological Systems: Towards a New Bionics? 102. Berlin, Heidelberg: Springer; 1993. p. 703-12.
9. Chakraborty A, Kar AK. Swarm intelligence: A review of algorithms. Nature-Inspired Computing and Optimization: Springer; 2017. p. 475-94.
10. Dorigo M, Caro GD. The ant colony optimization meta-heuristic. In: Corne D, Dorigo M, Glover F, Dasgupta D, Moscato P, Poli R, et al. editors. New ideas in optimization: McGraw-Hill Ltd., England, UK; 1999. p. 11-32.
11. Kennedy J, Eberhar R. Particle swarm optimization. *Proceedings of* International Conference on Neural Networks. 1995;4:1942–8.
12. Karaboga D, Basturk B, editors. Artificial Bee Colony (ABC) Optimization Algorithm for Solving Constrained Optimization Problems. Foundations of Fuzzy Logic and Soft Computing: 12th International Fuzzy Systems Association World Congress, Cancun, Mexico, June 18-21, 2007 Proceedings; 2007; Berlin, Heidelberg: Springer Berlin Heidelberg.Karaboga2007
13. Chu S-C, Tsai P-W, Pan J-S. Cat swarm optimization. Pacific Rim International Conference on Artificial Intelligence. 2006:854-8.
14. Yang X-S. Firefly Algorithms for Multimodal Optimization. In: Watanabe O, Zeugmann T. editors. Stochastic Algorithms: Foundations and Applications, SAGA 2009. 5792. Berlin, Heidelberg: Springer; 2009. p. 169-78.Yang2009
15. Yang X-S. A new metaheuristic bat-inspired algorithm. In: González JR, Pelta DA, Cruz C, Terrazas G, Krasnogor N. editors. Nature inspired cooperative strategies for optimization. Berlin, Heidelberg: Springer; 2010. p. 65-74.
16. Mirjalili S, Mirjalili SM, Lewis A. Grey wolf optimizer. Advances in Engineering Software ADV ENG SOFTW. 2014;69:46-61.
17. Yang X-S. Swarm-based metaheuristic algorithms and no-free-lunch theorems. In: Rafael P, Heitor S. editors. Theory and New Applications of Swarm Intelligence 9. Janeza Trdine: InTech; 2012. p. 1-16.
18. Dorigo M, Gambardella LM. Ant colony system: a cooperative learning approach to the traveling salesman problem. IEEE T EVOLUT COMPUT . 1997;1(1):53-66.
19. Mirjalili S, Gandomi AH. Chaotic gravitational constants for the gravitational search algorithm. Appl. Soft Comput. . 2017;53:407-19.
20. Sagban R, Ku-Mahamud KR, Abu Bakar MS. Reactive max-min ant system with recursive local search and its application to TSP and QAP. INTELL AUTOM **SOFT** CO. 2017;23(1):127-34.
21. Shan X, Liu K, Sun P-L. Modified bat algorithm based on lévy flight and opposition based learning. Sci. Program. 2016;2016.
22. Singh N, Singh S. A Modified Mean Gray Wolf Optimization Approach for Benchmark and Biomedical Problems. EVOL BIOINFORM. 2017;13:1–28.
23. Strumberger I, Bacanin N, Tuba M. Enhanced firefly algorithm for constrained numerical optimization. conference Congress on Evolutionary Computation. 2017:2120-7.
24. Rashedi E, Nezamabadi-Pour H, Saryazdi S. GSA: a gravitational search algorithm. Information sciences. Inf Sci. 2009;179(13):2232-48.
25. Adekanmbi O, Green P. Conceptual comparison of population based metaheuristics for engineering problems. The Scientific World Journal. 2015;2015.
26. Mirjalili S, Lewis A. The whale optimization algorithm. Adv Eng Softw 2016;95:51-67.
27. Mirjalili S, Gandomi AH, Mirjalili SZ, Saremi S, Faris H, Mirjalili SM. Salp swarm algorithm: a bio-inspired optimizer for engineering design problems. Adv Eng Softw . 2017;114:163-91.
28. Alobaedy MM, Ku-Mahamud KR, editors. Scheduling jobs in computational grid using hybrid ACS and GA approach. Computing, Communications and IT Applications Conference; 2014 20-22 October 2014; Beijing, China.
29. Cuevas E, Zaldívar D, Pérez-Cisneros M. Advances in Metaheuristics Algorithms: Methods and Applications: Springer; 2018.
30. Jallad J, Mekhilef S, Mokhlis H, Laghari J, Badran O. Application of hybrid meta-heuristic Techniques for optimal load shedding planning and operation in an islanded distribution network integrated with distributed generation. Energies. 2018;11(5):1134.
31. Oliva D, Hinojosa S, Demeshko M, editors. Engineering applications of metaheuristics*: an introduction*. Journal of Physics: Conference Series; 2017: IOP Publishing.
32. Slowik A, Kwasnicka H. Nature Inspired Methods and Their Industry Applications—Swarm Intelligence Algorithms. **IEEE** T IND INFORM. 2018;14(3):1004-15.

33. Wolpert DH, Macready WG. No free lunch theorems for optimization. IEEE T EVOLUT COMPUT. 1997;1(1):67-82.
34. Bhattacharyya S. Handbook of Research on Swarm Intelligence in Engineering. Hershey, PA: IGI Global; 2015.
35. Talbi E-G. Metaheuristics: from design to implementation. Hoboken, NJ: John Wiley & Sons; 2009.
36. Rochat Y, Taillard ÉD. Probabilistic diversification and intensification in local search for vehicle routing. J. Heuristics . 1995;1(1):147-67.
37. Glover F. Tabu search—part I. ORSA Journal on computing. 1989;1(3):190-206.
38. Jaeggi DM, Parks GT, Kipouros T, Clarkson PJ. The development of a multi-objective Tabu Search algorithm for continuous optimisation problems. Eur J Oper Res. 2008;185(3):1192-212.
39. Lozano M, García-Martínez C. Hybrid metaheuristics with evolutionary algorithms specializing in intensification and diversification: Overview and progress report. COMPUT OPER RES. 2010;37(3):481-97.
40. Sengupta RN, Gupta A, Dutta J. Decision Sciences: Theory and Practice. Boca Raton, FL: CRC Press; 2016. 1026 p.
41. Homayouni SM, Fontes DB. Metaheuristics for Maritime Operations. Hoboken, NJ: John Wiley & Sons; 2018.
42. Arin A, Rabadi G. Memory and learning in metaheuristics. Artificial Intelligence, Evolutionary Computing and Metaheuristics. 427. Berlin, Heidelberg: Springer; 2013. p. 435-76.
43. Sagban R, Mahamud KRK, Bakar MSA, editors. Reactive memory model for ant colony optimization and its application to TSP. IEEE International Conference on Control System, Computing and Engineering; 2014; Malaysia, Penang: .
44. Glover F, Kochenberger GA. Handbook of metaheuristics. New York, USA: Kluwer Academic; 2003.
45. Xie X-F, Wang Z-J. Cooperative group optimization with ants (CGO-AS): Leverage optimization with mixed individual and social learning. Appl Soft Comput.. 2017;50:223-34.
46. Stanger-Hall KF, Lloyd JE, Hillis DM. Phylogeny of North American fireflies (Coleoptera: Lampyridae): implications for the evolution of light signals. Mol Phylogenet Evol. 2007;45(1):33-49.
47. Khan WA, Hamadneh NN, Tilahun SL, Ngnotchouye JM. A review and comparative study of firefly algorithm and its modified versions. Optimization Algorithms-Methods and Applications: InTech; 2016. p. 281-313.
48. Abdelaziz A, Mekhamer S, Badr M, Algabalawy M. The firefly metaheuristic algorithms: Developments and applications. IEEJ 2015;6(7):1945-52.
49. Kazemzadeh-Parsi M. A modified firefly algorithm for engineering design optimization problems. IJSTM. 2014;38(M2):403-21.
50. Kiełkowicz K, Grela D. Modified Bat Algorithm for Nonlinear Optimization. IJCSNS. 2016:46-50.

تصنيف استخدام الذاكرة في التجريبيات المرتكزة على فئة ذكاء السرب

شيماء أكرم ياسر **كو روحانا كو محمود**

كلية الحاسبات، جامعة اوتارا، ماليزيا.

الخلاصة:

التجريبيات (metaheuristics) تحت فئة ذكاء السرب (swarm intelligence) اثبتت فعاليتها وأصبحت أساليب شائعة لحل مشاكل التحسين المختلفة. يمكن تصنيف التجريبيات، بناءً على استخدام الذاكرة ، الى خوارزميات مع ذاكرة وتلك بدون ذاكرة. يؤدي عدم وجود ذاكرة في بعض التجريبيات إلى فقدان المعلومات التي تم الحصول عليها في التكرارات السابقة. تميل التجريبيات إلى الانحراف عن المجالات الواعدة لمساحات البحث التي ستؤدي إلى حلول غير مثالية. تهدف هذه الورقة إلى مراجعة استخدام الذاكرة وتأثيرها على أداء أهم التجريبيات المرتكزة على ذكاء السرب. تم إجراء التحقيق على التجريبيات المرتكزة على على ذكاء السرب ، واستخدام الذاكرة و التجريبيات بدون ذاكرة ، وخصائص الذاكرة والذاكرة في التجريبيات المرتكزة على ذكاء السرب. تم تحليل المعلومات والمراجع لاستخراج المعلومات الأساسية وتعيينها في الأقسام الفرعية ذات الصلة. تم فحص ما مجموعه 50 مرجعًا تتعلق بدراسات استخدام الذاكرة من عام 2003 إلى عام 2018 ، وتبين أن استخدام الذاكرة ضروري للغاية لزيادة فعالية التجريبيات من خلال الاستفادة من تجاربها السابقة الناجحة. لذلك تعتبر الذاكرة في التجريبيات واحدة من العناصر الأساسية الفعالة للتجريبيات المتقدمة. كما تم تسليط الضوء على مشاكل في استخدام الذاكرة. نتائج هذه المراجعة مفيدة للباحثين في تطوير تجريبيات فعالة ، من خلال الأخذ بنظر الاعتبار استخدام الذاكرة.

الكلمات المفتاحية: تحسين شامل، استخدام الذاكرة، التجريبيات المستوحاة من الطبيعة، خوارزمية التحسين، خبرة البحث.