Songklanakarin J. Sci. Technol.
39 (5), 641-650, Sep - Oct. 2017

http://www.sjst.psu.ac.th

Review Article

Bat algorithm in discrete optimization: A review of recent applications

Wanatchapong Kongkaew*

Department of Industrial Engineering, Faculty of Engineering,
Prince of Songkla University, Hat Yai, Songkhla, 90112 Thailand

Received: 30 March 2016; Revised: 21 June 2016; Accepted: 16 August 2016

Abstract

Bat algorithm (BA), inspired by the foraging behavior of microbats, has become a powerful swarm intelligence method for solving optimization problems over continuous and discrete spaces. Nowadays, it has been successfully applied to solve problems in almost all areas of optimization, as well as engineering practices. Due to the limited applications in discrete structure, this paper carries out an updating review on recent applications of BA for discrete optimization problems. The solution mapping procedures are explained how to convert design variables between continuous domain and discrete domain in order to activate BA to solve discrete problems. To enhance the capability and applicability of BA, other combinatorial problems have been suggested and the potential ways for modification and hybridization are provided as well. The results demonstrated that BA is a promising nature-inspired metaheuristic algorithm for solving a variety of combinatorial problems. Furthermore, it has some significant advantages over other existing algorithms.

Keywords: swarm intelligence, bat algorithm, combinatorial optimization, modification and hybridization, review

1. Introduction

Discrete optimization or combinatorial optimization is a subset of mathematical optimization that concerns problems of finding an optimal solution in a finite or countably infinite set of feasible solutions, by minimizing or maximizing some objective function (Korte & Vygen, 2012). It is also one of the most studied and researched areas in scheduling, logistics and supply chain management, artificial intelligence, and other applications. Some well-known problems referred to NP-hard combinatorial optimization problems include traveling salesman problem (Saji & Riffi, 2016), vehicle routing problem (Zhou *et al*., 2013; Zhou *et al*., 2016), and so forth (see Korte & Vygen, 2012). Owing to the complexity of NP-hard problem, in spite of the existence of diverse approaches in the literature, an implementation of new metaheuristics still becomes an important challenge for the scientific community (Talbi, 2009).

Many modern optimization metaheuristic algorithms have been developed on swarm intelligence-based, which is inspired from nature (Yang, 2010b). The ways to design these algorithms are diverse; however, they can be classified into three sources of inspiration: physical-, social-, and biological systems. Nowadays, many new nature-inspired metaheuristic algorithms have been proposed in the literature. One of them is bat algorithm (BA) that is based on the foraging feature of microbats (Yang, 2010a). The BA should be interesting because it has a combination of major advantages of existing successful algorithms, such as particle swarm optimization (PSO), harmony search (HS), and simulated annealing (SA), under appropriate conditions. It employs not only a frequency-tuning technique to increase the diversity of solutions in the population but also the automatic mechanism to balance exploration and exploitation during the search process, by varying between pulse emission rates and loudness of bats. The BA becomes a simple and efficient algorithm to solve many continuous optimization problems (Yang & Gandomi, 2012); however, it is still found few published works successfully applied to solve discrete problems (Chawla & Duhan, 2015; Fister, 2013; Yang & He, 2013). Therefore, this article

* Corresponding author.
Email address: wanatchapong.k@psu.ac.th

aims to timely update a recent advance of BA applications only on combinatorial optimization problems that have been reported on international scientific databases. This paper also provides a useful guidance of further research on BA not only to apply for other combinatorial problems and also to enhance the search performance. This work should be carried out because BA seems to have advantages over existing metaheuristic algorithms due to simple concept and structure, ease of use, speed of obtaining solutions, and robustness. Moreover, the academic publications dealing with BA applications for combinatorial problems are still limited.

The rest of this paper is organized as follows. Section 2 provides a brief description of the original BA and a connection of BA to discrete problems is discussed in Section 3. Section 4 highlights the applications of BA to combinatorial problems. Next, Section 5 discusses on the advantage and disadvantage of BA and the direction for further research. Finally, Section 6 draws the conclusion of this paper.

2. Original Bat-Inspired Algorithm

Bat algorithm is a recent swarm-intelligence-based metaheuristic algorithm. It has been developed by Yang in 2010, inspired by the foraging behavior of microbats with varying pulse emission rate and loudness (Yang, 2010a). The three idealized rules of developing the BA are: all bats use the sound wave (ultrasound) reflection to forage for prey's location whilst flying in the dark; bats fly randomly to forage for prey according to velocity v_i at position x_i, frequency q_i and loudness L_i; the loudness varies from a large (positive) value L_0 to a minimum constant value $L_{\min}$.

In order to reach the vicinity of target, each bat is randomly assigned a frequency q_i of emitted pulse drawn uniformly from an interval $[q_{\min}, q_{\max}]$, and can automatically adjust the frequency within the same range. The pulse emission rate r_i can also be adjusted in an interval [0,1], where 0 denotes no pulse at all, and 1 denotes the maximum pulse emission rate. Given a virtual bat and a position updating strategy of its position x_i and velocity v_i in a D-dimensional search space, the new solution $x_i^{(t)}$, frequency $q_i^{(t)}$, and velocity $v_i^{(t)}$ (of each bat in the population) at generation t are generated by

$$q_i^{(t)} = q_{\min} + \left(q_{\max} - q_{\min}\right)\beta \tag{1}$$

$$v_i^{(t+1)} = v_i^{(t)} + \left(x_i^{(t)} - x_{GBest}\right)q_i^{(t)} \tag{2}$$

$$x_i^{(t+1)} = x_i^{(t)} + v_i^{(t+1)} \tag{3}$$

where $\beta \in [0,1]$ is a uniformly distributed random value over an interval [0,1], and x_{GBest} denotes the current global best solution obtained (by comparing all the solutions among all the n bats after the fitness evaluation process).

After updating the bats' velocities and their positions, the local search part is performed only if a generated random number is greater than the pulse emission rate r_i. A solution is then selected among the current best solutions to generate locally a new position for each bat using a random walk, which is given by

$$x_{new}^{(t)} = x_{GBest} + \varepsilon L^{(t)} \tag{4}$$

where $\varepsilon \in [0,1]$ is a uniformly distributed random number, and $L^{(t)} = \left\langle L_i^{(t)} \right\rangle$ denotes the average loudness value computed over the n bats at generation t. According to the selection procedure, a new solution will be accepted if a uniform random number is less than the current loudness L_i and the fitness value of current solution $f(x_i)$ is better than that of global best solution $f(x_{GBest})$.

In order to achieve the balance between exploration and exploitation during the search process, loudness $L^{(t)}$ and pulse emission rate r_i should be updated only if the candidate solution is improved as the iterations proceed. They are updated by

$$L_i^{(t+1)} = \alpha L_i^{(t)}, \ r_i^{(t)} = r_i^{(0)}\left[1 - \exp(-\gamma t)\right], \tag{5}$$

where α and γ are predefined constants. In general, once a bat has found its prey's location, loudness and pulse rate will be decreased and increased, respectively. The whole process of BA will be repeated until some pre-specified stopping criterion is satisfied. The operational steps of the standard BA proposed by Yang (2010a) can be summarized as the pseudocode in Figure 1.

3. Applying BA to Discrete Problem

Due to the BA originally designed for continuous optimization, it is inapplicable when the set of feasible solutions for problem is discrete. To remedy this obstacle, a solution mapping procedure has been utilized to enable BA in discrete structure, depending on the characteristic of a given combinatorial problem. Here, three procedures of solution representation: permutation-based (continuous and discrete versions) and binary-based (binary version), have been briefly explained. Among them, two versions have been illustrated through an example in Figure 2.

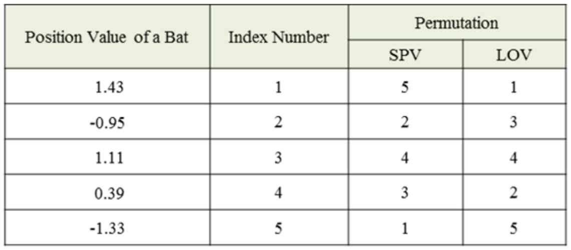
First, as in Figure 2(a), the permutation-based method with continuous version is related to a smallest position value (SPV) rule (Tasgetiren *et al.*, 2004) and a largest-order-value (LOV) rule (Qian *et al.*, 2009), which are based on random key representation. For example, according to SPV rule, the smallest position value is -1.33 corresponding to the (dimension) index number 5. Consequently, the position with index number 5 is attached 1 for the permutation. The second smallest value is -0.95 corresponding to the index number 2, so the position with index number 2 is attached 2, and so on. Under the SPV mechanism, the suggested permutation is (5, 2, 4, 3, 1). In other word, SPV rule is performed by sorting the indices of elements in ascending order according to the position values.

Then the permutation indicates a tentative solution under the mechanism of BA to the discrete problem, such as a job sequence for the single machine scheduling problem and a complete tour for the traveling salesman problem.

Algorithm Standard Bat Algorithm

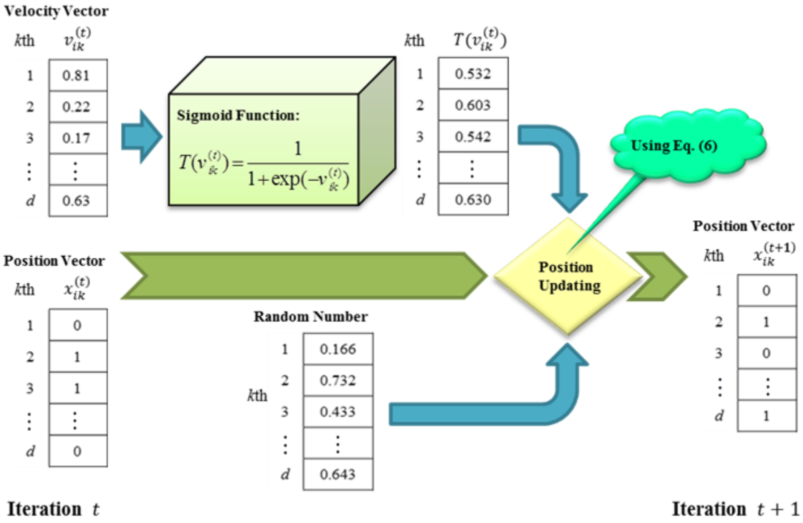
```
Input: Bat population $x_i = (x_{i1}, x_{i2}, \dots, x_{id})^T$, for $i = 1, 2, \dots, NP$,
       velocity $v_i$, pulse rates $r_i$, loudness $L_i$, pulse frequency $q_i$ at $x_i$, and
       maximum number of generations $Max\_Gen$.
Output: The best solution $x_{GBest}$ and its corresponding fitness value $f(x_{GBest})$
  Initial bat $x_i$, $i = 1, 2, \dots, NP$
  Evaluate fitness for each bat $f(x_i)$
  while ($t < Max\_Gen$)
     Generate new solutions by adjusting frequency, and updating
     velocities and locations/solutions
     if (rand $> r_i$)
        Select a solution among the best solutions
        Generate a local solution around the selected best solution
     end if
     Generate a new solution by flying randomly
     if (rand $< L_i$ and $f(x_i) < f(x_{GBest})$)
        Accept the new solutions
        Increase $r_i$ and decrease $L_i$
     end if
     Rank the bats and update the current best solution $x_{GBest}$
     Increase the generation number $t$
  end while
```

Figure 1. Pseudocode of the standard bat algorithm.



Position Value of a Bat	Index Number	Permutation	
		SPV	LOV
1.43	1	5	1
-0.95	2	2	3
1.11	3	4	4
0.39	4	3	2
-1.33	5	1	5

(a) Permutation-based representation



(b) Binary-based representation

Figure 2. Example of solution representations for discrete problem.

Second, discrete version, there are some literatures that invent a suitable representation by encoding each bat in population directly as a permutation of integer numbers, i.e., one bat would be represented as $x_i = (5,2,4,3,1)$. However, the structure of standard BA is required for modification on formulations for the bat movement and the local search part; see Zhou *et al*. (2013), Luo *et al*. (2014), Saji and Riffi (2016), and Osaba *et al*. (2016) for more details.

Finally, binary version, the binary-based representation introduced by Kennedy and Eberhart (1997) is applied to activate transformation between continuous space and binary space (dealing with only two numbers: "0" or "1") via a sigmoid transfer function and a position updating strategy. A sigmoid function is used to convert all values in velocity vector of each bat to probability values. These values are then utilized for updating positions of a corresponding bat with the rules:

$$x_{ik}^{(t+1)} = \begin{cases} 0 & \text{If } rand < T(v_{ik}^{(t+1)}) \\ 1 & \text{If } rand \geq T(v_{ik}^{(t+1)}) \end{cases} \tag{6}$$

where k denote the kth dimension of position (or velocity) vector, and *rand* is a uniformly distributed random number over [0,1). The position updating strategy employs formula in Eq. (6) instead of one in Eq. (3) to enable BA in binary space due to a link of procedures between position and velocity of a bat.

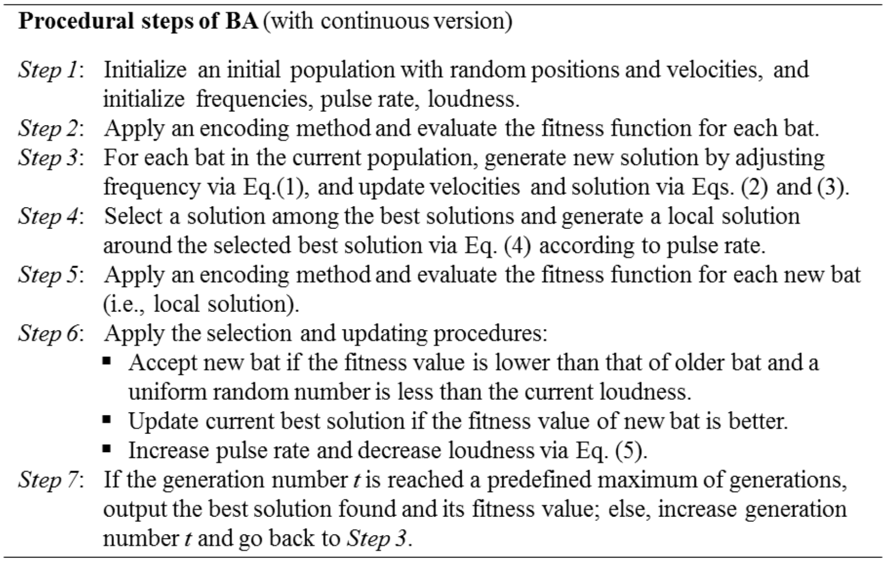
For example, as in Figure 2(b), the velocity value 0.17 at iteration t in the 3rd dimension can be transformed through sigmoid function, producing a probability value 0.542. Given a random number 0.433 in the 3rd dimension, Equation (6) is then proceeded to update the position value of a bat. Consequently, the position value in the 3rd dimension is switched from "1" to "0" for iteration t+1. Furthermore, Mirjalili and Lewis (2013) provided an investigation on performances of existing transfer functions and updating processes to map solution into binary structure over the last two decades. To execute with BA, Figure 3 demonstrates the steps of BA when applying to a specific combinatorial problem.

4. Applications of Bat Algorithm in Discrete Optimization

This section highlights the applications of BA to combinatorial problems which have revealed little evidence, if any, on researches published since its establishment in 2010. These applications can be categorized into five groups based on problem characteristics: scheduling, routing, allocation, facility layout design, and combination of multi-problem. The numbers of articles published in each group are 9, 5, 2, 4, and 1 paper, respectively. Moreover, according to literatures, there are three versions of BA based on encoding scheme: continuous, discrete, and binary (with 11, 8, and 2 papers, respectively), to solve discrete problems.

4.1 Scheduling

The bat algorithm based scheduling tool, introduced by Musikapun and Pongcharoen (2012), is implemented to find optimal sequence of operations to manufacture components for multi-stage multi-machine multi-product scheduling problem. Reported results show that the tuned BA works better than uncalibrated one. For the multiprocessor scheduling problem (MSP), Malakooti *et al*. (2012) implemented BA to solve either single- or bi-objective MSP in order to minimize makespan and tardiness. Results indicate that BA outperforms genetic algorithm (GA) and some well-known dispatching rules. Marichelvam and Prabaharam (2012) applied BA to minimize both makespan and mean flow time on the hybrid flow shop (HFS) scheduling problem. Based



Procedural steps of BA (with continuous version)

Step 1: Initialize an initial population with random positions and velocities, and initialize frequencies, pulse rate, loudness.
Step 2: Apply an encoding method and evaluate the fitness function for each bat.
Step 3: For each bat in the current population, generate new solution by adjusting frequency via Eq.(1), and update velocities and solution via Eqs. (2) and (3).
Step 4: Select a solution among the best solutions and generate a local solution around the selected best solution via Eq. (4) according to pulse rate.
Step 5: Apply an encoding method and evaluate the fitness function for each new bat (i.e., local solution).
Step 6: Apply the selection and updating procedures:

- Accept new bat if the fitness value is lower than that of older bat and a uniform random number is less than the current loudness.
- Update current best solution if the fitness value of new bat is better.
- Increase pulse rate and decrease loudness via Eq. (5).

Step 7: If the generation number *t* is reached a predefined maximum of generations, output the best solution found and its fitness value; else, increase generation number *t* and go back to *Step 3*.

Figure 3. Procedure of BA for combinatorial problem.

on realistic data gathered from a furniture manufacturing company, BA outperforms GA, PSO, SA, ant colony optimization (ACO), and artificial immune system. Following that, Marichelvam *et al.* (2013) proposed BA to minimize makespan on the multistage HFS problem. Computational experiments on different benchmark problems indicate that BA is an efficient method and performs superior to GA, PSO, and parallel greedy algorithm.

Later, Xie *et al.* (2013) introduced a differential Lévy-Flights bat algorithm to minimize makespan for permutation flow shops problem (PFSP). In the algorithm, a combination of the Nawaz, Enscore, and Ham (NEH) heuristic and random initialization is utilized to construct an initial population, and a virtual population neighborhoods search is applied to avoid local traps. Reported results show the effectiveness of BA. Luo *et al.* (2014) presented a discrete BA for solving PFSP, where the whole problem is divided into many sub-problems and each has been solved by NEH heuristic. Two local search strategies, moreover, are carried out and controlled by pulse rate in local search part. Next, Tosun and Marichelvam (2016) presented a hybridization of BA and local search techniques to minimize makespan for PFSP. Reported results indicate that it performs more efficient than many existing metaheuristic algorithms.

The hybridization of BA with a guided population and two-swap local search for single machine scheduling problem is addressed in my previously published work (Kongkaew, 2015). Experimental results show that hybrid BA with both techniques outperforms the rest of all possible designs. To minimize total cost of aircrafts landing deviation from the target time, Xie *et al.* (2013) implemented a hybrid BA (HBA) with the swap- and loop subsequence inserting methods to find optimal schedules of landing aircrafts in case of multiple runways. Reported results show that HBA is very effective and comparative.

4.2 Routing

Zhou *et al.* (2013) and Zhou *et al.* (2016) presented hybrid BA with path relinking (HBA-PR), by integrating greedy randomized adaptive search procedure, path relinking, random subsequence and random single-point local searches, to solve capacitated vehicle routing problem (CVRP). Experimental results on different benchmark instances indicate that HBA-PR is effective and outperforms other existing algorithms. Ochoa *et al.* (2013) applied BA to determine optimal route of a transportation fleet based on vehicle routing problem for a caravan range community. Following that, Taha *et al.* (2015) proposed an adapted bat algorithm (ABA) for CVRP which allows to generate high-diversity population and to attain the balance of exploration and exploitation in the search. Computational results on selected benchmark instances indicate that ABA is effective and outperforms GA.

Saji and Riffi (2016) presented a discrete BA with 2-opt mechanism for solving the symmetric traveling salesman problem (TSP). Later, motivated by this work, Osaba *et al.* (2016) adopted Hamming distance to the velocity equation for bat movement and extend an additional mechanism of 3-opt moves into local search part for solving symmetric and asymmetric TSP. Reported results indicate that the discrete BA works better than basic BA, and it outperforms GA, island based distributed GA, evolutionary SA, discrete firefly algorithm, and imperialist competitive algorithm.

4.3 Allocation

For the multidimensional knapsack problem, Sabba and Chikhi (2014) introduced a binary BA in which the sigmoid function is employed to convert a solution to binary numbers. Experimental results on benchmark instances and comparison with other bio-inspired algorithms indicate that it is a very promising algorithm. Talafuse and Pohl (2016) applied BA to solve the redundancy allocation problem (RAP) for series-parallel system with component mixing. Computational results show that BA is competitive with ACO for the RAP.

4.4 Facility layout design

Dapa *et al.* (2013) introduced BA for designing machine layout in order to minimize total material handling distance when non-identical machines have been arranged in a multiple-row layout with limited area of manufacturing shop floor. Results show that BA well performs on benchmark datasets taken from literature. To design the cellular manufacturing system, Parika *et al.* (2013) applied BA to minimize total number of movement between cells for the cell formation design problem with a consideration on routing flexibility. Later, Büyüksaatçı and Baray (2014) and Büyüksaatçı (2015) proposed BA to minimize sum of products of the flow in single row facility layout problem.

4.5 Combination of multi-problem

Sadeghi *et al.* (2014) presented a calibrated hybrid bat algorithm (called C-HBA in this article) with PSO and three local searches to optimize a bi-objective inventory model incorporated RAP and TSP for the vendor managed inventory policy in a three-echelon supply chain. Experimental results and comparison study show that C-HBA works better than standard BA and GA. Table 1 provides a summary of study on some applications of BA for combinatorial problems. This table focuses on three main topics: version of BA based on encoding scheme, hybridization, and features of metaheuristics.

5. Why the BA Should Be of Interest for Researchers

5.1 Comparison with other methods

Despite of the existence of many well-known metaheuristics, it is still necessary to seek a new method due

Table 1. Review study of some BA applications for combinatorial problems.

Year	Author	Version of BA			Hybrid	Techniques used for hybridization	Features of Metaheuristics			
		Continuous	Discrete	Binary			Parameter Tuning	Re-start mechanism	Memory structures	Multi-neighborhood
2012	Musikapun and Pongcharoen	✓					✓			
2012	Marichelvam and Prabaharam	✓					✓			
2013	Xie *et al*.		✓		✓	• Swap and loop subsequence inserting local searches	✓			✓
2013	Zhou *et al*.		✓		✓	• Greedy randomized adaptive search procedure • Path-relinking procedure • 2-opt local search • Single-point local search (swap and insertion) • Sub-sequence insertion and inversion	✓	✓	✓	✓
2013	Xie *et al*.	✓			✓	• Virtual population neighborhood search • NEH heuristic		✓		
2014	Sabba and Chikhi			✓	✓	• Sigmoid function-based local search • Random search strategy				✓
2014	Sadeghi *et al*.	✓			✓	• Particle swarm optimization • swap, inversion, and reversion local searches	✓			✓
2014	Luo *et al*.		✓		✓	• NEH heuristic • Single-point local search (swap, insertion, inversion, and crossover) • Sub-sequence insertion and inversion	✓	✓		✓
2015	Saji and Riffi		✓		✓	• 2-opt local search	✓			
2015	Kongkaew	✓			✓	• Two-swap local search • Weighted modified due date method				
2016	Talafuse and Pohl			✓			✓			
2016	Osaba *et al*.		✓		✓	• 2-opt and 3-opt local searches		✓		✓

to some disadvantages of existing algorithms. This section discusses advantages and disadvantages of BA by comparing with existing methods. In fact, the BA is a recent nature-inspired metaheuristic algorithm that integrates key advantages of PSO, HS, and SA. In other words, PSO and HS can be considered as special cases of BA under appropriate conditions, and the parameter α in Equation 5 acts as cooling factor of cooling schedule in SA (Yang, 2010a). In comparison to direct method (i.e., mixed-integer programming, MIP), BA does not guarantee optimality while MIP does. However, BA can solve near-optimal solutions quickly with less computational effort than MIP due to a property of metaheuristic method.

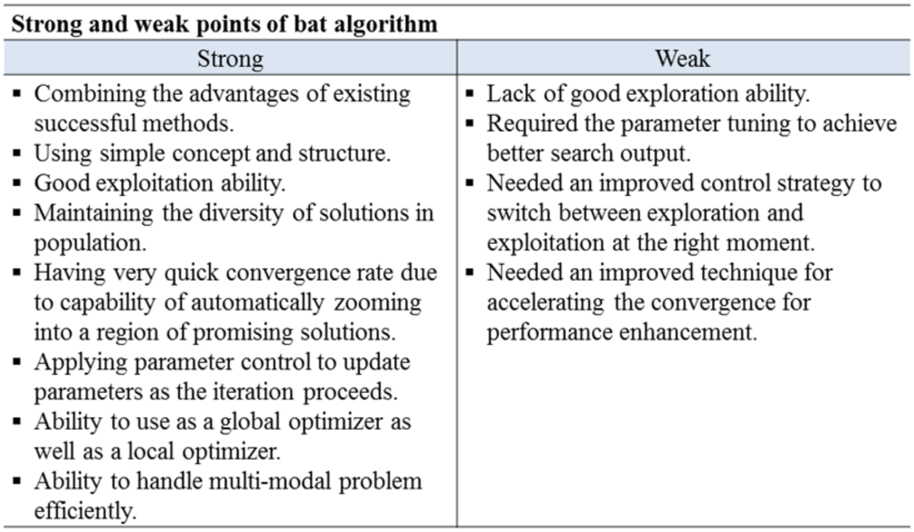
Comparing with other metaheuristic algorithms, BA has more diversity of solutions than SA due to the population characteristics. As reported in Yang (2010a) and Yang and Gandomi (2012), BA performs better than GA and it seems that BA has equivalent or possibly better performance to differential evolution (DE) and PSO. Moreover, all parameters of BA can be varied as the iteration proceeds while the parameter values used in standard GA, PSO, and DE versions are fixed. For some combinatorial problems, the BA is potentially better accuracy and efficiency than SA, GA, PSO, ACO, and firefly algorithm (Marichelvam & Prabaharam, 2012; Marichelvam *et al*., 2013; Osaba *et al*., 2016; Talafuse & Pohl, 2016).

Furthermore, BA has a capability of automatically zooming into a promising solution region that permits local intensive exploitation; other metaheuristic algorithms do not. Although the BA has performed very efficient in exploitation, it seems to have relatively poor in exploration ability. This is because BA has no crossover operation (unlike GA and DE); consequently, BA maintains the members of the whole population through the search procedure. Based on knowledge gained from literatures, the strong and weak points of BA can be concluded as in Figure 4.

5.2 Further research directions

This section discusses some possible recommendations for further research of bat algorithm not only on the applications to other combinatorial optimization problems but also on the ways to enhance search capability. Nowadays, BAs have been successfully applied in many optimization problems, as well as practical engineering problems in several fields (Yang, 2010a). As seen from this review, BAs are suitable for various types of combinatorial problems (either single- or multi-objective formulation) including but not limited to the following domains: scheduling, routing, allocation, facility layout design, and combination of multi-problem. Due to the limited applications for combinatorial optimization, the addition of specific problems related to this field is depicted in Figure 5.

According to hybridization discussed in Puchinger and Raidl (2005), there are two main categories that BA can incorporate with either exact or other metaheuristic algorithms: collaborative and integrative combinations. The first one is that BA exchanges information, but it is not a part of other algorithm. The latter means that BA can be embedded as a sub-module of another method. Based on collaborative combination, the other methods (as in Figure 5) can be used as initialization of population in the BA. For example, Sadeghi *et al*. (2014) applied PSO to generate the first population of BA for permutation problems; consequently, hybrid BA performs better than standard BA in terms of solution quality and computation time. Combining with exact methods (such as MIP-based method), the exact method may be used to divide the search space of problem into sub-problem, and each sub-problem is then solved by BA for a solution. On the other hand, BA is applied to find near-optimal solutions of problem in the first part of hybrid algorithm while the exact method is applied to second part as post-optimization procedure of solutions generated by the first.



Strong and weak points of bat algorithm

Strong	Weak
▪ Combining the advantages of existing successful methods. ▪ Using simple concept and structure. ▪ Good exploitation ability. ▪ Maintaining the diversity of solutions in population. ▪ Having very quick convergence rate due to capability of automatically zooming into a region of promising solutions. ▪ Applying parameter control to update parameters as the iteration proceeds. ▪ Ability to use as a global optimizer as well as a local optimizer. ▪ Ability to handle multi-modal problem efficiently.	▪ Lack of good exploration ability. ▪ Required the parameter tuning to achieve better search output. ▪ Needed an improved control strategy to switch between exploration and exploitation at the right moment. ▪ Needed an improved technique for accelerating the convergence for performance enhancement.

Figure 4. Strong and weak analysis of bat algorithm.

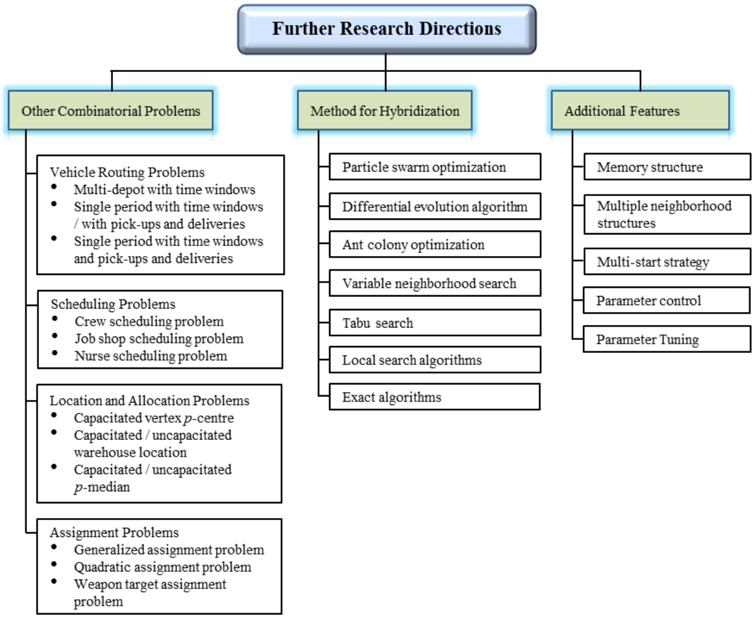

Further Research Directions
Other Combinatorial Problems
Vehicle Routing Problems
Multi-depot with time windows
Single period with time windows / with pick-ups and deliveries
Single period with time windows and pick-ups and deliveries
Scheduling Problems
Crew scheduling problem
Job shop scheduling problem
Nurse scheduling problem
Location and Allocation Problems
Capacitated vertex p-centre
Capacitated / uncapacitated warehouse location
Capacitated / uncapacitated p-median
Assignment Problems
Generalized assignment problem
Quadratic assignment problem
Weapon target assignment problem
Method for Hybridization
Particle swarm optimization
Differential evolution algorithm
Ant colony optimization
Variable neighborhood search
Tabu search
Local search algorithms
Exact algorithms
Additional Features
Memory structure
Multiple neighborhood structures
Multi-start strategy
Parameter control
Parameter Tuning

Figure 5. A guidance for some possible extensions of BA for combinatorial problems.

As the integrative combination, BA can be hybridized with another approach as either master or slave algorithms. Osaba *et al*. (2016), for example, applied either 2-opt or 3-opt local search to be operators in the position updating mechanism and local search part of BA for the TSP. Reported results indicate that hybrid BA outperforms standard BA. Moreover, the entire operation of DE algorithm can be used as a mechanism of population initialization (Step 1 in Figure 3) or local search part (Step 4) of BA. Due to the convergence acceleration, some part of DE (especially mutation operator) can be also employed to explore new solution between any two bats in population or new search space after completing local search part. On the other hand, Steps 3 and 4 of the BA are possibly used as an alternative to the DE mutation operator.

Last but not least, in the manner of additional features, a memory structure of Tabu search can be embedded into Step 6 of the BA in order to prevent cycling of revisited solutions during the search. In order to achieve a global optimum, multiple neighborhood structures permit to change neighborhoods of current solution if and only if an improvement is obtained. This feature can be applied to BA in many ways. A variable neighborhood search (VNS), for instance, deploys this concept and it may be used as a local search method in Step 4 instead of one in Equation 4 or applied to explore population individuals in Step 3 in order to crawl into the search space.

6. Conclusions

Bat algorithm (BA) has become a powerful nature-inspired metaheuristic algorithm for many continuous and discrete optimization problems. Nowadays, BA has widely expanded its implementation in almost every area of optimization and engineering applications. This paper provides an updating literature review on applications of BA, focusing on combinatorial optimization. According to the past literature review, it found that the number of academic articles related to applications of BA with discrete structure is quite limited even though BA is well-known as a new, simple, and efficient algorithm for continuous optimization tasks.

In this paper, a broad overview of the BA involves a wide range of its variants and applications for combinatorial optimization in recent literatures since its establishment in 2010. To tackle discrete problems using BA, three encoding schemes are concisely described on their mechanisms. Moreover, some features in the metaheuristics design that are used in recent variants of BA have been classified and summarized from some selected literatures. The advantages and disadvantages of BA over other existing methods have been also discussed.

Furthermore, some possible directions for further research of applying BA not only on other combinatorial problems but also on techniques for improving search performance (i.e., the hybridization with other existing algorithms, and the modification with the addition of several features) are recommended and discussed as well. This review has shown that BA is a promising swarm intelligence technique for combinatorial optimization due to its simplicity, flexibility, efficiency, and robustness. It has some significant advantages over other existing algorithms and can be applicable for solving a variety of combinatorial problems.

Acknowledgements

The author would like to gratefully thank for constructive comments and suggestions of editors and anonymous referees that are helpful to improve the manuscript.

References

Büyüksaatçı, S., & Baray, A. (2014). A comparison of three nature-inspired metaheuristics for the facility layout problem. In M. Papadrakakis, M. G. Karlaftis, & N. D. Lagaros (Eds.), *Proceedings of the 1st International Conference on Engineering and Applied Sciences Optimization (OPTI 2014)* (pp. 1032-1043). Kos Island, Greece.

Büyüksaatçı, S. (2015). Bat algorithm application for the single row facility layout problem. In X. - S. Yang (Ed.), *Studies in computational intelligence: Vol. 585. Recent advances in swarm intelligence and evolutionary computation* (pp. 101-120). Cham, Switzerland: Springer International Publishing.

Chawla, M., & Duhan, M. (2015). Bat algorithm: A survey of the state-of-the-art. *Applied Artificial Intelligence, 29*(6), 617-634.

Dapa, K., Loreungthup, P., Vitayasak, S., & Pongcharoen, P. (2013). Bat algorithm, genetic algorithm and shuffled frog leaping algorithm for designing machine layout. In S. Romanna, P. Lingras, C. Sombattheera, & A. Krishna (Eds.), *Lecture notes in computer science: Vol. 8271. Multi-disciplinary trends in artificial intelligence* (pp. 59-68). Berlin, Germany: Springer-Verlag.

Fister, I., Jr. (2013). *A comprehensive review of bat algorithms and their hybridization* (Master's thesis, University of Maribor, Maribor, Slovenia).

Kennedy, J., & Eberhart, R. (1997). A discrete binary version of the particle swarm algorithm. *Proceedings of the International Conference on Systems, Man, and Cybernetics*, 4104-4108.

Kongkaew, W. (2015). Solving the single machine total weighted tardiness problem using bat-inspired algorithm. *Proceedings of the 2015 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, 265-269.

Korte, B., & Vygen, J. (2012). *Combinatorial optimization: Theory and algorithms* (5th ed.). Berlin, Germany: Springer-Verlag.

Luo, Q., Zhou, Y., Xie, J., Ma, M., & Li, L. (2014). Discrete bat algorithm for optimal problem of permutation flow shop scheduling. *The Scientific World Journal, 2014*, 1-15.

Malakooti, B., Kim, H., & Sheikh, S. (2012). Bat intelligence search with application to multi-objective multiprocessor scheduling optimization. *International Journal of Advanced Manufacturing Technology, 60*(9-12), 1071-1086.

Marichelvam, M. K., & Prabaharam, T. (2012). A bat algorithm for realistic hybrid flowshop scheduling problems to minimize makespan and mean flow time. *ICTACT Journal on Soft Computing, 3*(1), 428-433.

Marichelvam, M. K., Prabaharan, T., Yang, X.-S., & Geetha, M. (2013). Solving hybrid flow shop scheduling problems using bat algorithm. *International Journal of Logistics Economics and Globalisation, 5*(1), 15-29.

Mirjalili, S., & Lewis, A. (2013). S-shaped versus v-shaped transfer functions for binary particle swarm optimization. *Swarm and Evolutionary Computation, 9*, 1-14.

Musikapun, P., & Pongcharoen, P. (2012). Solving multi-stage multi-machine multi-product scheduling problem using bat algorithm. *Proceedings of the 2nd International Conference on Management and Artificial Intelligence*, 98-102.

Ochoa, A., Margain, L., Arreola, J., De Luna, A., García, G., Soto, E., . . . Scarandangotti, V., (2013). Improved solution based on bat algorithm to vehicle routing problem in a caravan range community. *Proceedings of the 13th International Conference on Hybrid Intelligent Systems*, 18-22.

Osaba, E., Yang, X.-S., Diaz, F., Lopez-Garcia, P., & Carballedo, R. (2016). An improved discrete bat algorithm for symmetric and asymmetric traveling salesman problems. *Engineering Applications of Artificial Intelligence, 48*, 59-71.

Parika, W., Seesuaysom, W., Vitayasak, S., & Pongcharoen, P. (2013). Bat algorithm for designing cell formation with a consideration of routing flexibility. *Proceedings of the 2013 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, 1353-1357

Puchinger, J., & Raidl, G. R. (2005). Combining metaheuristics and exact algorithms in combinatorial optimization: A survey and classification. In J. Mira, & J. R. Álvarez (Eds.), *Lecture notes in computer science: Vol. 3562. Artificial intelligence and knowledge engineering applications: A bioinspired approach* (pp. 41-53). Berlin, Germany: Springer-Verlag.

Qian, B., Wang, L., Hu, R., Huang, D. X., & Wang, X. (2009). A DE-based approach to no-wait flow-shop scheduling. *Computers & Industrial Engineering, 57*(3), 787-805.

Sabba, S., & Chikhi, S. (2014). A discrete binary version of bat algorithm for multidimensional knapsack problem. *International Journal of Bio-Inspired Computation, 6*(2), 140-152.

Sadeghi, J., Mousavi, S. M., Niaki, S. T. A., & Sadeghi, S. (2014). Optimizing a bi-objective inventory model of a three-echelon supply chain using a tuned hybrid bat algorithm. *Transportation Research Part E: Logistics and Transportation Review, 70*, 274-292.

Saji, Y., & Riffi, M. E. (2016). A novel discrete bat algorithm for solving the travelling salesman problem. *Neural Computing and Applications, 27*(7), 1853-1866.

Taha, A., Hachimi, M., & Moudden, A. (2015). Adapted bat algorithm for capacitated vehicle routing problem. *International Review on Computers and Software, 10* (6), 610-619.

Talafuse, T. P., & Pohl, E. A. (2016). A bat algorithm for the redundancy allocation problem. *Engineering Optimization, 48*(5), 900-910.

Tasgetiren, M. F., Sevkli, M., Liang, Y.-C., & Gencyilmaz, G. (2004). Particle swarm optimization algorithm for single machine total weighted tardiness problem. *Proceedings of the 2004 Congress on Evolutionary Computation (CEC2004)*, 1412-1419.

Talbi, E.-G. (2009). *Metaheuristics: From design to implementation*. Hoboken, NJ: John Wiley & Sons.

Tosun, Ö., & Marichelvam, M. K. (2016). Hybrid bat algorithm for flow shop scheduling problems. *International Journal of Mathematics in Operational Research, 9*(1), 125-138.

Xie, J., Zhou, Y., & Tang, Z. (2013). Differential Lévy-Flights bat algorithm for minimization makespan in permutation flow shops. In D.-S. Huang, K.-H. Jo, Y.-Q. Zhou, & K. Han (Eds.), *Lecture notes in computer science: Vol. 7996. Intelligent computing theories and technology* (pp. 179-188). Berlin, Germany: Springer-Verlag.

Xie, J., Zhou, Y., & Zheng, H. (2013). A hybrid metaheuristic for multiple runways aircraft landing problem based on bat algorithm. *Journal of Applied Mathematics, 2013*, 1-8.

Yang, X.-S. (2010a). A new metaheuristic bat-inspired algorithm. In J. R. González, D. A. Pelta, C. Cruz, G. Terrazas, & N. Krasnogor (Eds.), *Studies in computational intelligence: Vol. 284. Nature inspired cooperative strategies for optimization (NICSO 2010)* (pp. 65-74). Berlin, Germany: Springer-Verlag.

Yang, X.-S. (2010b). *Nature-inspired metaheuristic algorithms*. Frome, England: Luniver Press.

Yang, X.-S., & Gandomi, A. H. (2012). Bat algorithm: A novel approach for global engineering optimization. *Engineering Computations, 29*(5), 464-483.

Yang, X.-S., & He, X. (2013). Bat algorithm: Literature review and applications. *International Journal of Bio-Inspired Computation, 5*(3), 141-149.

Zhou, Y., Luo, Q., Xie, J., & Zheng, H. (2016). A hybrid bat algorithm with path relinking for the capacitated vehicle routing problem. In X.-S. Yang, G. Bekdaş, & S. M. Nigdeli (Eds.), *Modeling and optimization in science and technologies: Vol. 7. Metaheuristics and optimization in civil engineering* (pp. 255-276). Cham, Switzerland: Springer International Publishing.

Zhou, Y., Xie, J., & Zheng, H. (2013). A hybrid bat algorithm with path relinking for capacitated vehicle routing problem. *Mathematical Problems in Engineering, 2013*, 1-10.