

Received July 15, 2021, accepted July 28, 2021, date of publication August 3, 2021, date of current version August 26, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3102154

Hybrid Bird Mating Optimizer With Single-Based Algorithms for Combinatorial Optimization Problems

ANAS ARRAM^{ID}, MASRI AYOB^{ID}, AND ALAA SULAIMAN^{ID}

Centre for Artificial Intelligence (CAIT), Data Mining and Optimization Research Group (DMO), Universiti Kebangsaan Malaysia, Bangi, Selangor 43600, Malaysia

Corresponding author: Anas Arram (anas.aram@gmail.com)

This work was supported in part by the Ministry of Higher Education Malaysia for under Grant TRGS/1/2019/UKM/01/4/1, and in part by Universiti Kebangsaan Malaysia under Grant DIP-2019-013.

ABSTRACT Bird mating optimizer (BMO) is a population-based metaheuristic that has been recently extended to solve combinatorial optimization problems. Even though the algorithm shows promising performance in solving combinatorial optimization problems, it suffers from slow convergence and poor efficiency which leads to poor solution quality for some problem instances. This is due to the limited capability of BMO in exploiting the search space and identifying more promising regions. Therefore, in this work we propose a hybrid BMO with five single-based metaheuristics: hill-climbing, late acceptance hill-climbing, simulated annealing, iterated greedy heuristic and variable iterated greedy heuristic. Each of these algorithms is used inside the BMO to exploit the search space, and improve the quality of solution generated from the BMO population. This work also compares which one of these five is better for hybridizing with BMO. The performance of these algorithms is tested on two combinatorial problems: travelling salesman problem and berth allocation problem. Experimental results demonstrate that the hybrid algorithm is superior to BMO when applied to both problems and it improved the BMO by 1.13% for BAP and by 4.13% for TSP. Furthermore, the hybrid algorithm is able to match the best-known results for most of the instances. In addition, the proposed hybrid approaches perform well over both tested domains and obtain competitive results when compared to the best-known results that have previously been presented in the scientific literature.

INDEX TERMS Bird mating optimizer, berth allocation problem, travelling salesman problem, combinatorial optimization, hill climbing, late-acceptance hill-climbing, simulated annealing, iterated greedy.

I. INTRODUCTION

Combinatorial optimization problems (COPs) arise in many areas such as computer science, operational research, electronic commerce, and artificial intelligence. COPs can be defined as a topic to find an optimal ordering or arrangement for given discrete variables [1], [2]. Examples of these problems are job scheduling, journey planning, university educational timetabling, travelling salesman, and berth allocation [3]. COPs are NP-hard which may not be suited for exact methods that aim to find the actual optimal solution. Instead, approximate algorithms that can find (near) optimal solution with reasonable time are more practical especially when the problem instance is sizable [4]. Approximate algorithms consist

of heuristic and metaheuristic algorithms in which heuristic is a problem-dependent algorithm that is designed to solve a specific problem, while metaheuristic can work in effectively across different problems. Therefore, metaheuristics are the better choice when solving several COPs. Examples of these algorithms include simulated annealing (SA) [5], hill-climbing (HC) [6], late acceptance hill-climbing (LAHC) [7], genetic algorithm (GA) [8], [23], harmony search [9], iterated greedy algorithm (IG) [10]–[12], ant colony optimization [13], [14] and particle swarm optimization [15].

Bird Mating Optimizer (BMO) is a natural inspired metaheuristic algorithm that was proposed by [16]. The BMO inspired by mating behavior of birds during mating season, in which each male bird attempts to mate with a female one to breed a new brood. BMO has some advantages as avoiding trap in local optima by using five mating strategies

The associate editor coordinating the review of this manuscript and approving it for publication was Jamshid Aghaei^{ID}.

to move through the search space. In addition, BMO has more capabilities in exploring and exploiting the search space than other mating algorithms such as genetic algorithm [16]. BMO shows a competitive result comparing to other evolutionary algorithms such as classical evolutionary programming, fast evolutionary programming, classical evolutionary strategies, fast evolutionary strategies, genetic algorithm, particle swarm optimization, and group search optimizer [16]. However, the BMO was originally proposed to solve continuous optimization problems, but in 2020 [17] proposed a discrete version of BMO for solving COPs. In this work, we will use this version of BMO as it shows an effective and promising performance for solving COPs.

However, even though BMO shows promising performance in solving optimization problems, it suffers from slow convergence and poor efficiency for some constraint problems. This is due to the limited capability of BMO in exploiting the search space and identifying more promising regions. For some complicated problems, BMO shows premature convergence or poor efficiency [18], [19]. Therefore, maintain the balance between exploration and exploitation ability of the BMO algorithm is needed. Some researchers have attempted to tackle these issues by conducting several improvement modifications. One of these modifications is hybridizing BMO with single-based metaheuristics [19], [18] to improve the exploitation ability of the algorithm. In this study, our aim is to hybrid the BMO with five single-based metaheuristics, HC, LAHC, SA, IG and variable IG, to improve the performance of BMO for solving COPs. The reason for choosing five single-based metaheuristics is to compare and investigate which of these five can lead to better performance when hybridizing it with BMO.

According to Talbi *et al.* [6], metaheuristic hybridization can be done by combining a metaheuristic with a complementary metaheuristic. In this model, there are two levels of hybridization: low-level and high-level. In the low-level hybridization, a given function (e.g., crossover or mutation in GA) of metaheuristic is replaced by another metaheuristic (e.g., single-based metaheuristic). While in the high-level, the different metaheuristics are self-contained and there is no relationship between their internal working. For each level, two hybridization mechanisms are available: relay and teamwork hybridization. In relay hybridization, a set of metaheuristics is applied one by one and each one is taking the output of the previous one as input. Whilst, teamwork hybridization represents a set of optimization models that cooperated as agents in parallel, and each agent searches in solution space [6].

In this work, we aim to use the low-level teamwork hybrid (LTH) model for COPs, where a population-based algorithm BMO will be hybridized with five single-based algorithms, respectively. In other words, each single-based algorithm will be improving the exploitation ability in BMO and identifying more promising solution regions. The single-based algorithm will be applied with a probability to every newly generated solution in the population of BMO to ensure that it's the local

optima. In addition, a comparative study will be conducted between these five hybrid models to figure out which one is more effective in finding more promising solutions.

The rest of the paper is organized as follows: Section 2 represents the related work. The basics of BMO, HC, LAHC, SA, IG, and VIG are presented in Section 3. Section 4 presents the details of the proposed hybridization approach. Section 5 summarizes and analyzes the experimental. Finally, the conclusion and details of future works are given in Section 6.

II. METHODS

The following subsections present the details of the BMO algorithm and the five single-based algorithms.

A. BIRD MATING OPTIMIZER

BMO is a population-based metaheuristic algorithm proposed by Askarzadeh [16] that mimics the mating behavior of bird species during mating season. The population is referred as a society and each member of the society represents a feasible solution and called bird. The new generated solution called brood. The society contains of two components: males and females. Males are classified into three components: monogamous, polygynous, and polyandrous. While females are with the most promising genes which contain two categories: parthenogenesis and polyandrous. In total, the algorithm uses five updating strategies to generate new solutions and they are explained below with details. Note here that we are using the BMO version that proposed by Arram *et al.* [20] which was proposed to solve COPs.

Parthenogenesis is the mating system in which the female bird can produce a brood without mating with a male. In this system, each female tries to produce her brood by modifying and changing her genes with a predefined rate. Each female bird in the parthenogenetic group produces a brood using simple non-improvement hill-climbing (NI-HC). The NI-HC algorithm starts with a solution and generates a new one by applying some changes to the genes and repeat this process until there is no improvement to the current solution. Applying this system requires a probability at every iteration to decide to apply this algorithm or to skip it. The pseudocode of this system is presented in Algorithm 1 [16], [17].

Algorithm 1: Pseudocode of the HC Algorithm

Input: x as the initial solution.

Calculate the initial cost function $f(x)$

Repeat

 Generate x' . //generate a candidate neighbor//

 Calculate the cost of the generated solution $f(x')$

if x' better than x then // x' is the generated solution//

 | $x = x'$

end if

Until termination condition is satisfied.

Output: final solution found x .

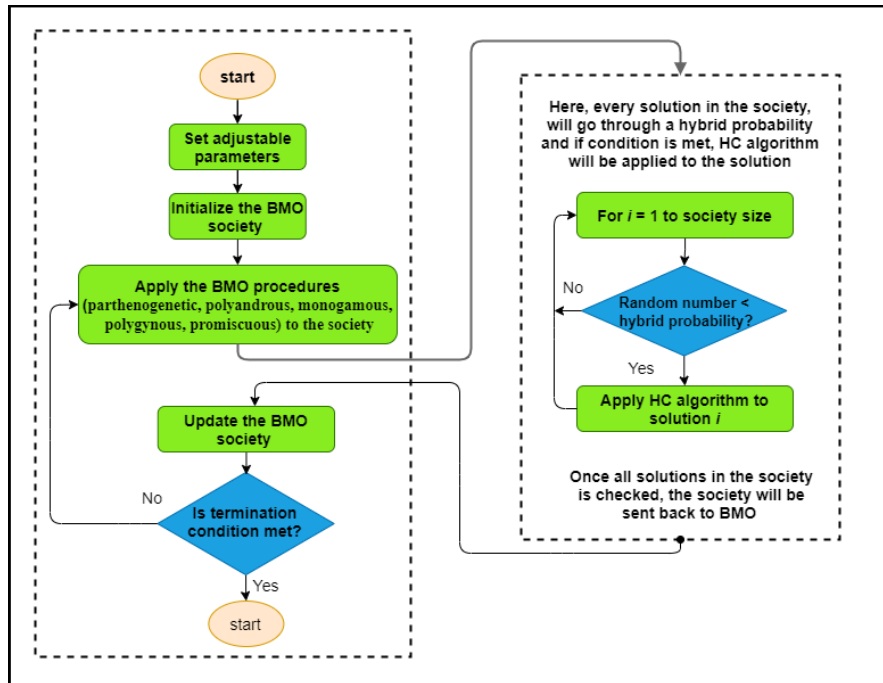


FIGURE 1. Flowchart of the hybrid BMO with HC.

Monogamy is a mating system in which a male tends to mate with only one female. Every male selects its mates from the female group (parthenogenesis and polyandrous) by evaluating the quality of the females using a probabilistic approach to select one of them. So female birds with good genes have a higher probability of being selected. For this system, the two-parent order crossover (OX) is used as a mating system between the male bird and its selected female [16], [17]. More details about OX are reported in [21]. The produced brood (new solution) from this mating system, will go through a mutation operator with a probability to improve the quality of the brood. Insertion operator will be used as a mutation operator where a gene is selected from the brood randomly and moved to another randomly selected position in the brood.

Polygamy is a mating system in which each polygynous male tries to produce a brood by mating with two or more females. The benefit of this multi-mating process is to produce a brood with better genes. In nature, a polygynous birds mate with several females to produce a number of broods, but in the BMO, only one brood results from this mating process, where the brood's genes are a combination of the male's and multiple females' genes. After selecting the females by using a selection mechanism, each male bird mates with his selected females. The multi-parent partially mapped crossover (MPPMX) is used in this system as a mating operator between males and several selected females. MPPMX was proposed by Ting *et al.* [22] as an extension of two-parent partially mapped crossover to multiparent crossover for better performance. The MPPMX applies four

main steps to generate a new brood: substring selection, substring exchange, mapping list determination and offspring legalization. More details of this crossover are reported in [22]. In addition, insertion mutation operator is applied to the produced brood with a probability to improve its quality [17].

Promiscuity is also another mating system in which one male mates with several females with an unstable relationships. This mating process indicates a chaotic social structure in which the male bird will never see the brood, and generally will not see the female for further mating activity. In promiscuity, the birds use a chaotic sequence method during the generations. However, each promiscuous bird behaves in the same way as a monogamous bird. Therefore, the OX is used in this mating process and insertion mutation operator will be applied with a probability [17].

Polyandry is the last mating system where a polyandrous female bird seeks to mate with more than two monogamous males. The female performs a selection mechanism to select the males. Then, each female bird mates with her selected male birds. MPPMX is used for this mating system as multiple birds will mate, and insertion mutation operator will be applied with a probability [17]. Note here that the mutation operator with a probability is applied to the four mating systems (Monogamy, Polygamy, Promiscuity and Polyandry) regardless of the resulting solution quality, and the aim of this is to maintain the diversity of the algorithm.

The following steps explain the BMO algorithm procedure as presented in [17]:

Step 1: Parameter initialization: initialize the BMO parameters.

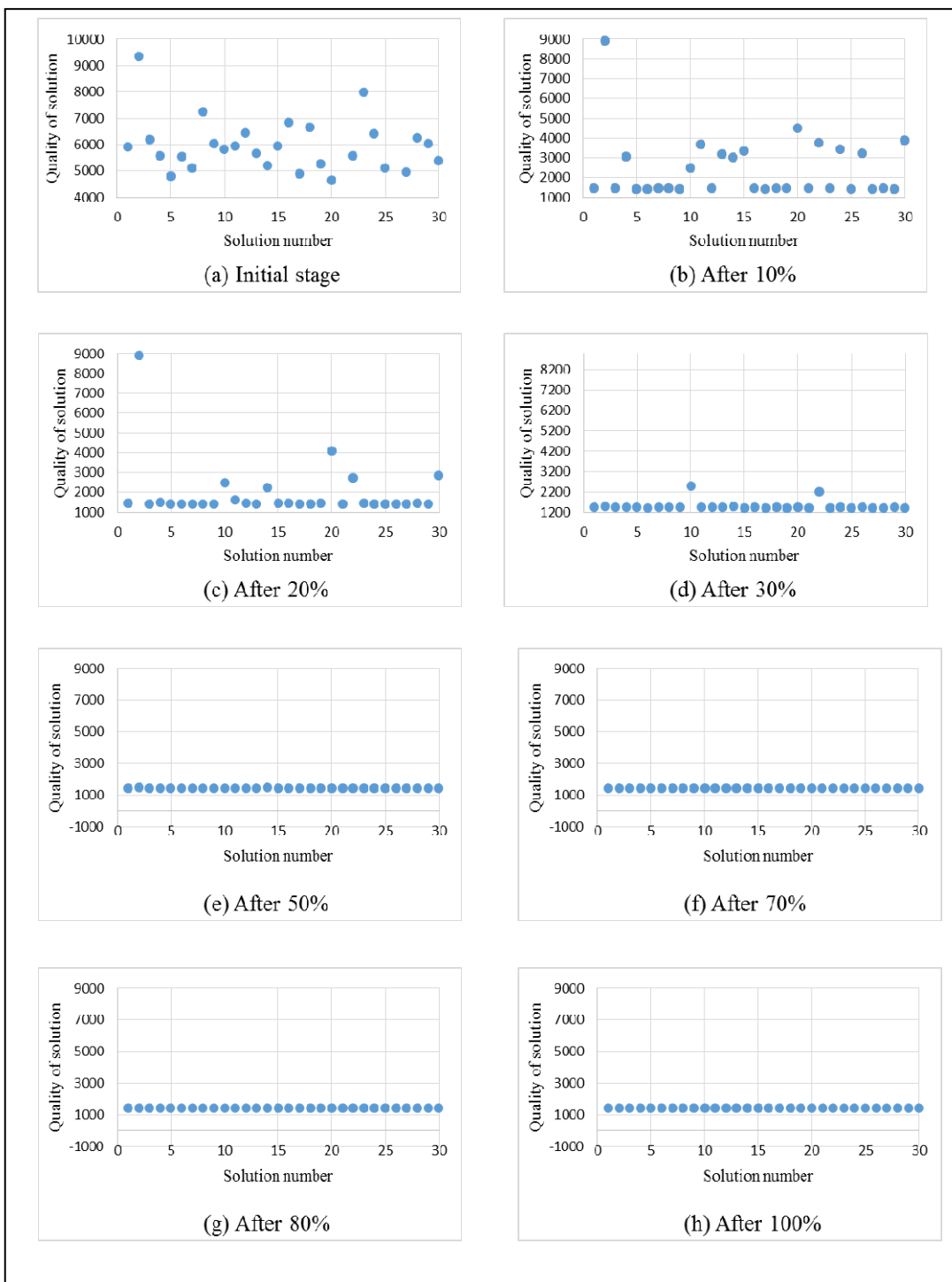


FIGURE 2. Diversity measurement of the hybrid BMO-HC on i01 BAP instance.

Step 2: Society initialization: initialize the population randomly with feasible birds (solutions). The initialization strategy is related to the problem domain.

Step 3: Society evaluation: calculate the quality of each bird using the objective function of the problem.

Step 4: Ranking: rank the birds in the society based on their quality in descending order.

Step 5: Classification: classify the birds in the society into five groups based on their quality, the classification is as follow: parthenogenetic, polyandrous, monogamous,

polygynous, and promiscuous birds. The percentage of each group is a parameter and defined in the parameters initialization step.

Step 6: Breeding: each bird produces a new brood using its own mating system.

Step 7: Replacement: If the quality of the brood is better than the quality of the bird, the brood replaces the bird. Otherwise, the bird remains in society, and the brood is removed.

Step 8: Termination condition: steps from 4 to 7 are repeated until a predetermined number of generations is performed.

Step 9: Report the best: select the bird with the best quality in society as the best solution.

The pseudocode of the DBMO is illustrated in Algorithm 2.

Algorithm 2: Pseudo-Code of LAHC Algorithm

Generate the initial solution x
Calculate initial cost function $f(x)$
Set the length of array L_{fa}
 Iteration $i = 0$
Repeat
 Construct a candidate solution x^*
 Calculate its cost function $f(x^*)$
 $v = i \bmod L_{fa}$
 if $f(x^*) \leq f_v$ or $f(x^*) \leq f(x)$
 | accept the candidate ($s = s^*$)
 else
 | reject the candidate ($x = x$)
 | Insert the current cost into the fitness array
 $f_v = f(x)$
 end $i = i + 1$
Until stopping criteria satisfied

B. HILL CLIMBING

Hill Climbing (HC) is one of the well-known local search methods that attempt to find better solution. The algorithm starts with an initial solution and improves it by generating a neighborhood solution. The current solution is replaced with a neighborhood solution if the quality of the neighborhood is better than the current, otherwise, the neighborhood is rejected, and HC begins a new iteration. The search process continues until the stopping criteria is satisfied [23], [24]. Algorithm 3 presents the pseudo-code of the algorithm.

C. LATE ACCEPTANCE HILL-CLIMBING

The late acceptance hill-climbing algorithm (LAHC) is an improved version of HC algorithm, which was proposed by Burke and Bykov [25]. The main idea of LAHC is to accept the non-improving solution when the quality of the new generated solution is better (or equal) than those that were recently accepted a few iterations before. In practice, LAHC starts from a given initial solution, and iteratively improves it by comparing the new candidate solution with the current one in order to accept or reject it. Hence, to apply the LAHC rule, the algorithm will create a list with a fixed length to

Algorithm 3: Pseudo-Code of SA Algorithm

Generate the initial solution $x = x_0$
Set the starting temperature $T = T_0$
Set Cooling rate β .
Set max number of trials at each temperature i_{max}
Repeat
 $i = 0$
 for $i = 0$ to i_{max} **do**
 Generate a random neighbor x'
 $\Delta E = f(x') - f(x)$
 if $\Delta E \leq 0$
 | $x = x'$ //accept the neighbor solution
 else
 Generate a random number between 0 and 1
 (r)
 if ($r < e^{-\frac{\Delta E}{T}}$)
 | $x = x'$
 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.

save the quality of recently visited solutions. If the quality of the new candidate solution is better than the quality value of the last element in the list, the candidate solution will be accepted as the current initial solution. Then, the element at the end of the list will be removed and the quality value of the newly accepted solution will be added to the beginning of the list [7]. Algorithm 4 presents the pseudo-code of the LAHC algorithm [7], [25].

D. SIMULATED ANNEALING

Simulated annealing (SA) was proposed by Kirkpatrick *et al.* [5], Černý [26] as an optimization algorithm for solving optimization problems. SA was proposed based on hill-climbing to overcome its problem of trapping in local optima. SA uses probability to accept worse solution and give more chances to explore the search space. The algorithm starts with a random initial solution, and at each iteration, a neighbor solution is generated using a predefined neighborhood structure. The new solution is evaluated using fitness function and will be accepted if it is better than the best one, otherwise, the worse solution is accepted with a probability that is determined by Boltzmann probability as in equation 1. In addition, T is a temperature that periodically decreases during the search process according to the cooling schedule [5], [27]. The pseudocode of SA is presented in Algorithm 5.

$$P = e^{-\Delta E/T} \quad (1)$$

where ΔE is the difference between the best solution and the generated one.

Algorithm 4: Pseudo-Code of IG algorithm

```

Generate the initial solution  $x$ 
Set the starting temperature  $T = T_0$ 
Repeat
  //destruction procedure
  for  $i = 1$  to  $d$  do
    //  $d$  is the destruction size
    remove one gene at random from  $x$  and add it in
     $x2$  //  $x2$  is a second array to store the removed
    genes from  $x$ 
  end
  //construction procedure
  for  $i = 1$  to  $d$  do
     $x'$  = best permutation obtained by inserting gene
     $x2_i$  in all possible position of  $x$ 
  End
  if the quality of  $x'$  is better than the  $x$  do
    |  $x = x'$  //accept the new solution
  Else
     $\Delta E = f(x') - f(x)$ 
    Generate a random number between 0 and 1 ( $r$ )
    if ( $r < e^{-\frac{\Delta E}{T}}$ )
      |  $x = x'$ 
    end
  end
  Update the best solution
Until Stopping criteria satisfied
Return best found solution.

```

E. ITERATED GREEDY ALGORITHM

Iterated greedy algorithm (IG) is a stochastic search method that was proposed by [10]. The IG generates a new solution using the idea of destruction and construction phases. In destruction phase, number of solution's genes d are randomly chosen and removed from the solution without repetition, so that two partial solutions will be resulted. The first, with the size d of genes, is donated as SR including the removed genes in the same order that they removed. The second, with the size $n-d$ of genes, is the original solution without the removed genes which is donated as SD . Next, the construction phase is employed to reinsert the removed genes into the solution. The NEH insertion heuristic is used as a constructive procedure to complete the solution. The first gene SR_1 is inserted into all possible $n-d+1$ positions in the destructed solution SD , which generates $n-d+1$ partial solutions. Among these $n-d+1$ generated partial solution, only the solution with best quality is chosen and kept for the next iteration. The second gene is then considered, and the process continues until SR is empty or final solution is obtained. Therefore, SD is again with the size of n [10]. In addition, the Boltzmann probability of SA is used here (see equation 1), where new generated solution is always accepted if it is better than the best, otherwise the worse one is accepted with the probability. But one different thing that is the temperature

Algorithm 5: Pseudo-Code of VIG Algorithm

```

Generate the initial solution  $x$ 
Set the starting temperature  $T = T_0$ 
Set  $d_{max} = n - 1$  //  $d_{max}$  is the maximum number for
destruction size.
 $n$  is the length of the solution
Repeat
   $d = 1$ 
  //destruction procedure
  for  $i = 1$  to  $d$  do
    remove one job at random from  $x$  and add it in
     $x2$ 
  end
  // construction procedure
  for  $i = 1$  to  $d$  do
     $x'$  = best permutation obtained by inserting gen
     $x2_i$  in all possible position of  $x$ 
  End
  if the quality of is better than the quality of  $x$ 
  |  $x =$  //accept the new solution
  Else
     $\Delta E = f(x') - f(x)$ 
    Generate a random number between 0 and 1 ( $r$ )
    if ( $r < e^{-\frac{\Delta E}{T}}$ )
      |  $x = x'$ 
      |  $d = d+1$ 
    else
      |  $d = d+1$ 
    end
  end
  Update the best solution
Until Stopping criteria satisfied
Return best found solution.

```

here is static, means there is no cooling schedule [10]. The pseudocode of IG is given in Algorithm 6 [10].

F. VARIABLE ITERATED GREEDY ALGORITHM

Variable iterated greedy algorithm (VIG) was proposed by Framinan and Leisten [28] to solve the permutation flowshop scheduling problem. The VIG is inspired from the variable neighborhood search (VNS) algorithm which presented in [29]. The VIG is developed by using the idea of neighborhood change of the VNS algorithm. The implementation of VIG is similar to IG but differs in the destruction parameter, where the maximum destruction size is fixed at $d_{max} = n-1$. The destruction size is initially set to $d = 1$. The current solution is destructed and reconstructed again with the variable size of d . Then, destruction size d is incremented by 1 (i.e., $d = d+1$), if the solution is not improved until $d_{max} = n-1$. Whenever the solution is improved in any destruction size, the destruction size is again set to $d = 1$ and the search starts from the beginning once again. The pseud-code of the VIG algorithm is given below [28].

Algorithm 6: Pseudocode of the Modified HC Algorithm

Input: x as the starting solution, mcf_p .
Improve = true.
Generate random number r between 0 and 1
If $r < \text{probability}$ **then**
 Repeat
 Generate $(N(x))$. //generate a candidate neighbor//
 if x' better than x **then** // x' is the generated solution//
 $x = x'$.
 improve = true.
 else
 improve = false.
 end
 Until improve = true.
Else $x = x$.
Output: final solution found x .

III. PROBLEMS DESCRIPTION

In this work, the performance of the proposed hybrid BMO is evaluated over two benchmark problems: The Travelling Salesman Problem (TSP) and Berth Allocation Problem (BAP). The following subsections describe these problems.

A. TRAVELLING SALESMAN PROBLEM

The TSP is a well-known combinatorial optimization problem that is classified as an NP-hard problem, which means that it may take an infeasible computational time to solve it [30]–[32]. The TSP can be described as a search for the shortest path route between a list of cities, that visits each city once and only once, and finally return to the origin city [33].

B. BERTH ALLOCATION PROBLEM

BAP is a NP-hard problem and known as a berth scheduling problem, where it aims to allocate berth space for vessels in container terminal to be served [34], [35]. The BAP can be classified into two models: the first one is discrete and continuous berth space, whilst the second is dynamic and static vessel arrival time. The berth space is considered as discrete if the quay is divided into a set of segments (berths), and continuous if not partitioned and vessels can berth anywhere along the quay. The vessel arrival time is considered as dynamic if vessels can arrive at any time during the container operations with planning arrival time, and static if all the vessels have to arrive in the port before the berth planning step begins. In this study, we focus on the discrete and dynamic version of the BAP [36]. This BAP deals with allocating vessels to berths in the port at the planned arrival time. More formally, the goal is to assign a berth for each vessel and a service time at the selected berth. The following assumptions are considered in the BAP [37]:

1) Each berth can serve only one vessel at a time;

2) Any vessel can be assigned to any berth with a given handling time taking into account that the handling time of a vessel can differ from one berth to another;

3) All vessels arrive at their berths before or after the berths' opening hours with a known arrival time;

4) When a vessel is moored in a berth, it remains there until all servicing activities have been completed.

The objective of the BAP is to minimize the overall waiting time of all the vessels that need to be serviced in the harbor, which is calculated as an objective function as follows [44]:

$$\sum_{i \in N} \sum_{k \in K} (T_i^k - a_i + P_i^k \sum_{j \in N \cup \{d\}} x_{ij}^k) \quad (2)$$

where:

- a_i : arrival time of vessel i
- K : set of berths,
- N : number of vessels that will arrive in the harbour,
- P_i^k : handling time of vessel i at berth k
- T_i^k : berthing time of vessel i at berth k
- x_{ij}^k : decision variable, $x_{ij}^k = 1$ if vessel j is serviced at berth k immediately after vessel i

IV. THE PROPOSED APPROACH

In the hybridization process, invoking single-based algorithms to the whole society (population) of BMO would be computationally expensive and also might cause a premature convergence in the BMO in the early stage of the search [38]. Therefore, the probability of hybridizing every member in the society is controlled based on a probability rate (hybrid probability (HP)) which is tuned based on preliminary experiments. As mentioned in Section 1, we are using a low-level teamwork hybrid, therefore the hybridization process is as follows. At each iteration, BMO generates a new society using its five mating systems (monogamous, polygynous, polyandrous, parthenogenesis, and polyandrous). Next, all birds in this new society will go through a hybrid probability rate, and any bird passes the probability rate, will go through a single-based algorithm, otherwise the bird will remain the same in the society. Finally, the society will go to society update step in BMO. The flowchart of the proposed hybrid algorithm is presented in Figure 1. The Figure 1 shows the process of hybridizing BMO with HC, however, the same process is applied to the remaining four single-based algorithms (LAHC, SA, IG, VIG).

Furthermore, the new solution that obtained from the single-based heuristic algorithms is generated as follows. For IG and VIG, the new solution is generated based on the greedy manner as explained in sections 3.4 and 3.5, this is because they are designed to generate a new solution based on that greedy manner. Whilst, the new solution generated from HC, LAHC and SA is generated using neighborhood structure. Hence, the type of the neighborhood structure depends on the problem domain. For BAP, the new solution is generated using swap operator, where two vessels at any berth are randomly selected, and their positions are exchanged [36]. For TSP, the neighborhood structure is generated using 2-opt

TABLE 1. Parameter settings of BMO.

parameter	Value for TSP	Value for BAP
Society size (SS)	30	30
Number of generations (NG)	4000	2000
Number of mates for polygynous and polyandrous birds (<i>nm</i>)	3	3
Mutation control factor (<i>mcf</i>)	0.1	0.1
The percentage of each group of birds in the society:		
Parthenogenetic	5%	5%
Polyandrous	5%	5%
Monogamous	50%	50%
Polygynous	30%	30%
promiscuous	10%	10%

neighborhood operator. The 2-opt operator selects two tours at random from the current solution and swaps the cities located at the end sections of the selected tours. This operator is different from the normal swap in the way that the swap operator only exchanges the places of the two selected cities.

V. EXPERIMENTS

The following sub-sections present the experimental setups, parameters tuning, and results.

VI. DATASETS

In order to assess the performance of the proposed BMO variants, the experiments are performed over two COPs: TSP and BAP. TSP benchmark includes 20 instances that range from 51 to 318 cities [13]. Whilst BAP involves 30 different instances, each instance has 30 vessels and 13 berths [35].

VII. PARAMETER SETTINGS

The proposed algorithms were implemented using Java Net-Beans IDE version 8.1 on a personal computer (Intel Pentium (R) Core i5 CPU at 3.40 GHz with 4 GB RAM), running a Windows 10 operating system (64-bit). All algorithms were executed over 30 independent runs with different random seeds for all instances of the problem domains.

The parameter settings of the BMO and single-based algorithms are set s follows:

A. BMO PARAMETER SETTINGS

Due to the impact of hybridizing single-based algorithms with BMO on its performance and diversity, the society size is carefully tuned. Therefore, the population size (PS) of the hybrid DBMO is fixed to 30 based on preliminary experiments for both TSP and BAP. The number of generations (NG) is also tuned based on preliminary experiments and set to 4000 and 2000 for TSP and BAP, respectively (see Table 10, Appendix A). The remaining parameters of BMO are fixed as suggested in [16] and presented in Table 1.

B. SINGLE-BASED PARAMETER SETTINGS

In this experiment, the stopping conditions of all single-based algorithms for both BAP and TSP are fixed based on the exe-

cutation time. Based on the preliminary test, the execution time is fixed to 0.2 sec for BAP and to 0.0015 sec * number of cities for TSP. Therefore, the results of the proposed algorithms can be fairly compared with each other. For the remaining parameters of the single-based algorithms, the settings are as follows:

- For HC parameters, HC has only one parameter, the maximum number of iterations. This parameter has been set based on the execution time as mentioned.
- LAHC has a list size (L) parameter, which is fixed to 10 for BAP and 50 for TSP based on preliminary experiments.
- SA has another two parameters in this work: Initial temperature (T) and cooling rate (β). The β is fixed at 0.85 and the T is fixed to be 50% of the value of the initial solution as suggested in [39], [40] for both BAP and TSP. Note that that loop will stop when the T reaches 0.
- IG has also a destruction size parameter (d), which is randomly selected in range of 4 to 7 as suggested in [41] for both TSP and BAP domains.
- VIG has another parameter which is the temperature level T . This parameter is fixed to 1000 for TSP based on preliminary tests, and to $0.05 \times TP/NBS$ for BAP as suggested in [41]. Where TP denotes total handling time (service time of a vessel on a berth), which is the summation of handling times of all vessels; NBS is the total number of allowable vessels for each berth, and is the number of vessels. For each berth, if the vessel can be served, it is considered an allowable vessel for the berth.

VIII. RESULTS AND DISCUSSION

All results obtained from the proposed hybrid BMO variants are presented in this section. The basic BMO are compared to it hybridized variants to assess the effect of hybridizing single-based algorithms to the basic BMO. To find out the best hybridized approach among the proposed five approaches, all approaches are compared together in one table. In addition, the best approach is compared to the state-of-the-art algorithms to verify the effectiveness of the proposed approach. Statistical analysis is also conducted to verify the significance of the best hybridized BMO variant over other variants.

In the subsequent tables, the results for each instance are presented as average (Avg.), standard deviation (Std), the best (Best), and percentage deviation (Gap%) with respect to the quality of solution produced by the compared algorithms. The best results are highlighted in bold and the Gap is calculated as follows:

$$\frac{BCA - BKS}{BKS} \quad (3)$$

where BCA show the best obtained from the compared algorithms and BKS stands for best-known solution in the literature.

Algorithm 7: Pseudocode of the DBMO Algorithm

Determine the society size (SS), maximum number of generations (gen_{max}), number of mates (nm) and mutation control factor (mcf).

Generate SS feasible birds.

for $t = 1$ to gen_{max} **do**

Rank the birds in ascending order based on their quality.

Classify the society into five groups: parthenogenetic polyandrous, monogamous, polygynous and promiscuous. //the classification is based on the quality of each bird.

for $i = 1$ to SS **do**

case parthenogenetic:

Produce new brood using NI-HC (Algorithm 4.3)

case polyandrous:

Select nm birds from the male group.

Produce new brood using MPPMX.

If $r < mcf$ then // r is a random number between 0 and 1//

Mutate the new brood using Insertion operator.

end

case monogamous:

Select one mate bird from females group.

Produce new brood using OX.

If $r < mcf$ then

Mutate the new brood using Insertion Operator

end

case polygynous:

Select nm birds from females group.

Produce new brood using MPPMX.

If $r < mcf$ then

Mutate the new brood using Insertion operator.

end

case promiscuous:

Select one mate bird from females group.

Produce new brood using OX.

If $r < mcf$ then

Mutate the new brood using Insertion operator.

end

end

end

Perform replacement stage: replace the new generated broods with their parents if they have better quality.

//update the society for the next generation

end

Return the best bird

A. COMPARISON BETWEEN BMO AND ALL PROPOSED HYBRID APPROACHES

In this section, we investigate the effectiveness of the hybrid BMO with single-based algorithms by comparing the results of the basic BMO with its hybrid approaches. The hybrid approaches with BMO are as follows: BMO-HC, BMO, LAHC, BMO-SA, BMO-IG, and BMO-VIG. In addition, we compare the five hybrid BMO approaches against each other to investigate which hybridized algorithm is better for each problem domain.

Table 2 and Table 11 (see Appendix A) present the results of BMO vs. BMO-HC, BMO-LAHC, BMO-SA, BMO-IG and BMO-VIG for the BAP. Whilst, Table 3 and Table 12 (see Appendix A) report the comparison results of BMO vs. BMO-HC, BMO-LAHC, BMO-SA, MO-IG and BMO-VIG for the TSP. Best results are shown in bold.

From Tables 1, 2 and 11 (see Appendix A), we can make the following observations:

- For BAP problem, across all instances, all hybrid BMO algorithms outperformed BMO not only in terms of percentage deviation, but also on average, standard deviation (see Table 2 and Table 10). We can observe that hybrid BMO with single-based algorithms enhanced the exploitation of BMO and produced better results. In comparison between all proposed hybrid BMO approaches, it's clear that BMO-HC performed better than BMO-LAHC, BMO-SA, BMO-IG and BMO-VIG not only in terms of percentage deviation, but also, on average results. Table 11 (see Appendix A) also shows that the standard deviation produced by BMO-HC is better than BMO-LAHC, BMO-SA, BMO-IG and BMO-VIG on 28 out of 30 instances. In addition, BMO-HC improved the BMO by 1.13% on the average of all instances in terms of solution quality (as shown in the last column in Table 2). On average computational time as can be seen in Table 12 (Appendix A), the values obtained from the BMO are less than those obtained from the hybrid BMOs. This is because the BMO suffers from poor exploitation ability, which led the algorithm to converge at an early stage of the search space. Thus, the best results are obtained with less time but with poor quality solution. In addition, the hybrid DBMOs require more time due to the use of SBH algorithms during the search, and this is still acceptable as the hybrid algorithms still run in a reasonable amount of time. Meta-heuristic techniques aim to provide a good quality solution with a reasonable amount of time, and this is what the hybrid BMO in this paper has achieved.
- For TSP, the results reported in Table 3 and Table 12 demonstrated that BMO-HC, BMO-LAHC and BMO-SA obtained better results for all tested

TABLE 2. Comparison between BMO and hybrid approaches for BAP over 30 independent runs (with respect to the quality of solutions).

Instances	BMO		BMO-HC		BMO-LAHC		BMO-SA		BMO-IG		BMO-VIG	
	Gap%	Avg	Gap%	Avg	Gap%	Avg	Gap%	Avg	Gap%	Avg	Gap%	Avg
i01	0.35	1428.1	0	1410	0	1411	0	1411	0	1411	0	1411
i02	0.08	1278.8	0	1261	0	1261	0	1261	0	1261	0	1261.2
i03	0	1139	0	1129.1	0	1129.7	0	1129.5	0	1134	0	1131.86
i04	0	1312.2	0	1302	0	1302	0	1302	0	1302	0	1302
i05	0	1220.1	0	1207	0	1207	0	1207	0	1207	0	1207
i06	0.87	1281.6	0	1261	0	1261.2	0	1261	0	1261.13	0	1261.13
i07	0.08	1287.7	0	1279.1	0	1279.7	0	1279.5	0	1279.62	0	1279.66
i08	0.23	1313.6	0	1299.03	0	1300.2	0	1299.7	0	1299.36	0	1299.66
i09	0.14	1460.5	0	1444.1	0	1444.3	0	1444.2	0	1444.26	0	1444.56
i10	0.08	1221.6	0	1213	0	1213	0	1213	0	1213	0	1213
i11	0.37	1382.1	0	1369	0	1371	0	1370	0	1370	0	1372
i12	0.68	1348.5	0	1329	0	1331	0	1330	0	1330.66	0.15	1330
i13	0.15	1378.2	0	1360	0	1360.2	0	1360	0	1360	0	1360
i14	0	1241.3	0	1233	0	1233	0	1233	0	1233	0	1233
i15	0	1304.1	0	1295	0	1295	0	1295	0	1295	0	1295
i16	0.59	1393.5	0	1365	0.07	1367	0	1366	0	1365.8	0	1366.66
i17	0	1289.2	0	1283	0	1283	0	1283	0	1283	0	1283
i18	0	1362.4	0	1345	0	1345	0	1345.1	0	1345.03	0	1345.4
i19	0.44	1387.5	0	1369	0.07	1372	0.07	1370	0.07	1370.66	0.07	1371
i20	0.23	1338.2	0	1329.5	0	1329.9	0.08	1330	0.08	1330.5	0.08	1330.8
i21	0.45	1354.9	0	1343	0	1346	0.07	1346	0.15	1345.73	0.15	1346
i22	0.23	1353.7	0	1326.3	0	1328.2	0	1327	0	1327.03	0	1327.76
i23	0.16	1288	0	1266.2	0	1268	0	1267	0	1267	0	1267.33
i24	0.08	1272.53	0	1260	0	1260.3	0	1260.2	0	1260	0	1260
i25	0.07	1393.27	0	1376.9	0.07	1377.9	0	1377	0	1377	0	1378.16
i26	0.3	1334.57	0	1318.6	0	1322	0	1320	0	1319	0	1319
i27	0.56	1271.53	0	1261	0	1261.6	0	1261	0	1261.4	0	1261.43
i28	0.22	1378.97	0	1360.3	0.07	1361	0	1361	0.07	1360.83	0.07	1361.43
i29	0.16	1291.93	0	1281.3	0.08	1281.9	0.08	1281.9	0	1281	0.08	1281.93
i30	0.3	1365.77	0	1348	0.3	1352	0	1350	0.22	1351.46	0.15	1352
Average	0.22	1322.4	0	1307.4	0.022	1308.5	0.01	1308	0.01	1308	0.02	1308

instances, in terms of the percentage deviation, average, standard deviation. Whilst, BMO-IG and BMO-VIG outperformed BMO on 17 and 13 out of 20 instances in terms of average results, respectively, and on 9 and 8 out of 20 instances in terms of standard deviation, respectively. Comparing the hybrid approaches against each other, we can see that BMO-SA outperformed BMO-HC, BMO-LAHC, BMO-IG and BMO-VIG on 19 out of 20 instances. On average and results, BMO-SA outperformed the four compared hybrid BMO approaches in 10 out of 20 instances (see Table 3). Table 13 (see Appendix A) also shows that BMO-SA produced better standard deviation compared to BMO-HC, BMO-LAHC, BMO-IG and BMO-VIG on only 5 out of 20 instances. In addition, BMO-SA improved the BMO by 4.13% on the average of all instances in terms of solution quality (as shown in the last column in Table 3). For computational time, as shown in Table 13, all hybrid BMO algorithms consumed more computational time compared to BMO. That is due to the early convergence in BMO and to the use of SBH in the hybrid approaches.

To verify whether the results are statistically different, we have conducted a pairwise comparison using Wilcoxon test with significant level of 0.05 for comparing BMO against hybrid approaches (Tables 15 and 15)7 and a multiple comparison statistical test to compare the hybrid approaches to each other (Tables 4-7). Table 15 and Table 16 show the *p*-values computed for the tested instances of the BAP and TSP, respectively. In this table, “+” indicates hybrid DBMO is statistically better than BMO (*p*-value < 0.05), “-” indicates hybrid DBMO outperformed by BMO (*p*-value > 0.05) and “=” indicates both hybrid BMO and BMO have the same performance (*p*-value = 0.05).

Based on *p*-value reported in Table 15 and Table 16, it can be concluded that BMO-HC, BMO-LAHC, BMO-SA, MO-IG and BMO-VIG results are statistically significant and better than from those produced by BMO across all instances of BAP problem domain (see Table 15). For the TSP, the *p*-values demonstrate that BMO-HC, BMO-LAHC and BMO-SA are statistically better than BMO on all tested instances, whilst BMO-IG and BMO-VIG are statistically better than BMO on 16 and 12 out of 6 instances, respectively (see Table 16).

TABLE 3. Comparison between BMO and hybrid approaches for TSP over 30 independent runs (with respect to the quality of solutions).

Instances	BMO		BMO-HC		BMO-LAHC		BMO-SA		BMO-IG		BMO-VIG	
	Gap%	Avg	Gap%	Avg	Gap%	Avg	Gap%	Avg	Gap%	Avg	Gap%	Avg
eil51	1.17	438.8	0	427	0	427	0	426	0	429	0	431
St70	0.89	692.5	0	690	0	677	0	676	0	680	0	687
eil76	22.86	675.7	0	542	0	543	0	538.5	0	549	0.37	551
Pr76	0.21	110211.1	0	110210	0	109647	0	104905	0	110989	0.33	111456
kroA100	1	21959.8	0	21310	0	21318	0	21338	0	22032	1.36	22857
kroB100	1.58	22963	0	22244	0	22258	0	22282	0.26	22953	2.46	23533
kroC100	0.16	21408.9	0	20815	0	20821	0	20861	0.63	21672	1.52	22125
kroD100	1.8	22161.3	0.07	21438	0	21408	0	21401	0	22190	3.21	23057
kroE100	1.38	22946.4	0	22204	0	22175	0	22233	1.17	22841	1.43	23216
rd100	1.24	8234.8	0	7948	0	7943	0	7944	0.63	8274	2.17	8403
eil101	5.09	675.7	0.16	638	0.32	640	0	634	2.07	656	2.07	661
lin105	0.23	14774.2	0	14392	0	14398	0	14417	0	15069	1.44	15333
ch130	3	6449.6	0.29	6175	0.05	6175	0	6162	1.62	6476	2.86	6666
ch150	4.14	7009.1	0.23	6598	0.23	6597	0	6572	4.12	7069	4.64	7232
kroA150	3.49	28134.1	0	26782	0	26774	0	26892	4.31	28850	7.51	29752
kroB150	2.42	27425.8	0.14	26373	0.14	26334	0	26365	5.36	28537	7.09	29318
D198	2.33	16334.4	0.3	15940	0.2	15965	0	15790	6.02	18456	6.87	18674
kroA200	4.39	31487.2	0.45	29768	0.29	29795	0.22	29790	10.3	1089.6	6.6	35163
kroB200	4.96	31627	0.57	29921	0.23	29920	0.35	30066	10.4	1270.1	14.3	35460
lin318	6.51	45764.3	2.94	43719	2.95	43814	2.46	43838	32.8	3459.7	30.8	65089
Average	3.4425	22068.7	0.25	21406.7	0.22	21381.4	0.15	21156.5	3.98	17177.1	5.08	23983.2

TABLE 4. The average rank of hybrid BMO algorithms obtained by Friedman, Aligned Friedman and Quade tests for BAP.

No.	Algorithm	Friedman	Aligned Friedman	QUADE
BAP	DBMO-HC	1.583	32.65	1.21-
	DBMO-SA.	2.95	65.28	2.83
	DBMO-IG	2.8	73.11	2.85
	DBMO-LAHC	3.9	103.7	4.14
	DBMO-VIG	3.766	102.75	3.95

TABLE 5. The average rank of hybrid BMO algorithms obtained by Friedman, Aligned Friedman and Quade tests for TSP.

No.	Algorithm	Friedman	Aligned Friedman	QUADE
TSP	DBMO-SA	1.95	30.75	1.89
	DBMO-HC	1.97	30.37	2.03
	DBMO-LAHC	2.07	30.37	2.07
	DBMO-IG	4	78.15	4
	DBMO-VIG	5	82.85	5

For multiple comparison between the hybrid approaches, Table 4 and Table 5 show the average ranking (the lower the better) computed by Friedman, Friedman Aligned and Quade tests for the hybrid BMO approaches (BMO-HC, BMO-LAHC, BMO-SA, MO-IG and BMO-VIG) for both BAP and TSP domains, respectively. The tables highlighted that BMO-HC and BMO-SA obtained the first rank out of five compared methods for BAP and TSP problem domains, respectively. The p -value computed through the statistics of each of the test considered are less than 0.05 and the Iman-Davenport

TABLE 6. The adjusted p -values computed by Friedman test for the compared methods for BAP instances.

Algorithm	Unadjusted P	P Holm	P Hochberg
DBMO-HC	0.96	1.605	0.96
DBMO-LAHC	0.802	1.605	0.960
DBMO-IG	0.00004	0.00012	0.00012
DBMO-VIG	0	0	0

Note: DBMO-SA is the controlling method

TABLE 7. The adjusted p -values computed by Friedman test for the compared methods for TSP instances.

Algorithm	Unadjusted P	P Holm	P Hochberg
DBMO-SA.	0.0008	0.0016	0.0016
DBMO-IG	0.0028	0.0028	0.0028
DBMO-LAHC	0	0	0
DBMO-VIG	0	0	0

Note: DBMO-HC is the controlling method

(<0.05) prove that there is a significant difference among the methods considered for both BAP and TSP. Therefore, post-hoc procedures are performed to detect the significant difference between all tested methods.

Table 6 shows the adjusted p -values of Holm and Hochberg statistical tests using the ranks computed by Friedman, Friedman Aligned and Quade tests, respectively where DBMO-HC is the controlling method. Table 6 demonstrate that BMO-HC outperforms other hybrid BMO approaches with a critical level of 0.05 (adjusted p -values < 0.05) for all tests

TABLE 8. Comparison between the proposed approaches and the state-of-the-art approaches for BAP in terms of best obtained solution (minimum cost).

<i>Ins.</i>	<i>BMO-HC</i>	<i>TS</i>	<i>CG</i>	<i>IG</i>	<i>SA</i>	<i>ALNS</i>
	<i>Best</i>	<i>Best</i>	<i>Best</i>	<i>Best</i>	<i>Best</i>	<i>Best</i>
i01	1409	1415	1409	1409	1409	1409
i02	1261	1263	1261	1261	1261	1261
i03	1129	1139	1129	1129	1129	1129
i04	1302	1303	1302	1302	1302	1302
i05	1207	1208	1207	1207	1207	1207
i06	1261	1262	1261	1261	1261	1261
i07	1279	1279	1279	1279	1279	1279
i08	1299	1299	1299	1299	1299	1299
i09	1444	1444	1444	1444	1444	1444
i10	1213	1213	1213	1213	1213	1213
i11	1368	1378	1369	1368	1368	1368
i12	1325	1325	1325	1325	1325	1325
i13	1360	1360	1360	1360	1360	1360
i14	1233	1233	1233	1233	1233	1233
i15	1295	1295	1295	1295	1295	1295
i16	1364	1375	1365	1364	1364	1364
i17	1283	1283	1283	1283	1283	1283
i18	1345	1346	1345	1345	1345	1345
i19	1367	1370	1367	1367	1367	1367
i20	1328	1328	1328	1328	1328	1328
i21	1341	1346	1341	1341	1341	1341
i22	1326	1332	1326	1326	1326	1326
i23	1266	1266	1266	1266	1266	1266
i24	1260	1261	1260	1260	1260	1260
i25	1376	1379	1376	1376	1376	1376
i26	1318	1330	1318	1318	1318	1318
i27	1261	1261	1261	1261	1261	1261
i28	1359	1365	1360	1359	1359	1359
i29	1280	1282	1280	1280	1280	1280
i30	1344	1351	1344	1344	1344	1344

TABLE 9. Comparison between the proposed approaches and the state-of-the-art approaches for TSP in terms of best obtained solution (minimum cost).

<i>Ins.</i>	<i>BMO-SA</i>	<i>GA</i>	<i>PSO</i>	<i>VTPSO</i>	<i>ABCSS</i>	<i>DSMO</i>
	<i>Best</i>	<i>Best</i>	<i>Best</i>	<i>Best</i>	<i>Best</i>	<i>Best</i>
eil51	426	5427.24	1627.32	429.51	428.98	428.86
St70	675	23976	5076	682.57	682.57	677.11
eil76	538	40037.96	23935.62	559.25	550.24	558.68
Pr76	108159	4154387	753868.2	109586.1	108879.7	108159.4
kroA100	21282	1531878	604834.4	21307.44	21299	21298.21
kroB100	22141	1411710	513671.2	22475.67	22229.71	22308
kroC100	20749	1581904	529929.5			
kroD100	21294	1624732	512972.5			
kroE100	22068	1653335	683446			
rd100	7910	567779.8	200755.8	8094.75	7944.32	8041.3
eil101	629	8931.8	13070.62	653.16	646.05	648.66
lin105	14379	1126019	314612.5	14581.58	14406.12	14383
ch130	6110	514828.6	253565			
ch150	6528	601751	345004.8			
kroA150	26524	3185267	1776578	27232.1	26981.98	27591.44
kroB150	26130	3118877	1631557	26579.73	26760.79	26601.94
D198	29368	4782579	2570287	30602.81	30701.86	30481.35
kroA200	35913.14	4893018	3407333	30767.52	31508.85	30716.5
kroB200	21303	2281788	1612874	16066.44	16270.22	15978.13
lin318	145420.3	10291221	11442816	44724.38	55744.52	44118.66

considered for BAP. Whilst for TSP, it can be seen in Table 7 that the BMO-SA outperforms BMO-IG and BMO-VIG with a critical level of 0.05 (adjusted p -values < 0.05). However, the results in Table 7 indicate that BMO-SA

does not outperform BMO-SA and BMO-LAHC (adjusted p -values > 0.10).

From the above comparison, we can conclude that hybridizing single-based algorithms with BMO enhances the

TABLE 10. Parameter tuning for BMO algorithm using 5 BAP and TSP instances based on best solution quality over 10 runs.

instances	Number of iterations - BAP				
	500	1000	1500	2000	3000
i01	1603	1516	1491	1479	1479
i10	1345	1270	1253	1251	1251
i15	1452	1358	1334	1336	1336
i20	1493	1407	1391	1376	1376
i25	1524	1457	1442	1429	1429

instances	Number of iterations - TSP				
	1000	2000	4000	6000	8000
eil51	453	454	451	451	451
lin105	17256	15794	15694	15694	15694
ch130	7865	6871	6737	6737	6737
kroA200	55390	39858	34193	34190	34190
lin318	140042	88411	62216	62212	62212

TABLE 11. Standard deviation comparison between BMO and hybrid approaches for BAP over 30 independent runs.

Instances	BMO	BMO-HC	BMO-LAHC	BMO-SA	BMO-IG	BMO-VIG
	<i>Std</i>	<i>Std</i>	<i>Std</i>	<i>Std</i>	<i>Std</i>	<i>Std</i>
i01	9.39	1.03	1.78	1.39	1.33	1.38
i02	8.88	0	0	0	0	0.69
i03	4.87	0.26	1.19	0.77	2.34	1.95
i04	7.35	0	0	0.19	0	0.19
i05	6.56	0	0	0	0	0
i06	4.75	0	0.49	0	0.74	0.74
i07	7.61	0.37	0.87	0.72	0.81	0.83
i08	5.89	0.19	1.14	1.14	0.62	1.02
i09	7.01	0.26	0.59	0.49	0.53	0.77
i10	5.16	0	0	0.19	0	0
i11	6.64	1.59	2.01	1.84	2.48	2.4
i12	8.33	2.48	2.39	2.39	2.26	1.86
i13	8.53	0	0.49	1.17	0	0
i14	6.63	0	0	0	0	0
i15	5.79	0	0	0	0	0
i16	12.99	1.36	1.9	2.02	1.64	1.78
i17	5.11	0	0	0	0	0
i18	7.52	0	0.67	0.45	0.19	0.87
i19	9.66	1.34	2.04	1.98	2.15	1.91
i20	6.89	0.72	1.13	0.96	1.07	1.23
i21	3.73	1.7	2.09	2.28	1.7	1.94
i22	12.26	0.83	1.7	1.52	1.16	1.56
i23	7.31	0.62	1.23	1.07	1.47	1.23
i24	6.12	0	0.64	0.45	0	0
i25	10.5	1.08	1.56	1.31	1.17	1.03
i26	6.44	1.7	2.33	2.17	1.75	2.12
i27	3.62	0	1.61	0	1.41	1.52
i28	13.18	0.93	1.35	1.53	1.33	1.22
i29	6.7	0.64	1.02	1.19	0.89	1.08
i30	7.43	2.29	2.24	3.44	3.12	3.66
Average	7.4	0.64	1.08	1.02	1.0	1.09

search process and the performance of BMO. It can be stated that BMO with single-based algorithm is a robust algorithm that balances between exploration and exploitation when searching for global optimum. The single-based algorithm improves the exploitation ability of BMO and

at the same time, it does not damage its exploration ability.

To show the hybrid algorithm improves the exploitation ability of the algorithm, we have conducted an analysis using one BAP instance as shown in Figure 2. The figure shows that

TABLE 12. Average computational time comparison between BMO and hybrid approaches for BAP over 30 independent runs (time in second).

Instances	BMO	BMO-HC	BMO-LAHC	BMO-SA	BMO-IG	BMO-VIG
	<i>Avg. Time(s)</i>	<i>Avg. Time(s)</i>	<i>Avg. Time(s)</i>	<i>Avg. Time(s)</i>	<i>Avg. Time(s)</i>	<i>Avg. Time(s)</i>
i01	2.31	26.82	29.29	34.51	27.04	31.93
i02	2.106	13.63	24.81	16.11	21.06	25.53
i03	2.265	22.02	20.98	28.48	37.03	41.3
i04	2.104	8.19	15.93	17.78	16.92	18.1
i05	2.113	6.81	13.24	9.28	3.44	4.78
i06	2.175	14.6	27.83	20.04	19.51	26.69
i07	2.158	17.58	11.18	15.18	13.61	13.3
i08	2.16	23.97	28.91	27.83	30.5	32.5
i09	2.099	15.7	22.39	28.23	28.66	30.66
i10	2.15	6.62	16.93	17.5	7.26	9.89
i11	2.1	31.81	31.79	32.86	41.57	37.4
i12	2.071	35.12	27.79	30.66	36.2	35.78
i13	2.228	13.97	26.3	22.34	10.32	13.12
i14	2.13	4.42	9.71	8.24	6.41	9.0
i15	2.145	5.02	9.21	9.04	7.93	13.34
i16	2.13	25.22	33.17	32.12	37.36	35.54
i17	2.035	2.93	4.47	5.0	1.2	1.62
i18	2.184	12.77	24.46	23.83	24.89	33.8
i19	2.204	34.87	37.31	31.46	38.68	37.34
i20	2.138	19.67	20.38	23.17	32.96	38.98
i21	2.107	36.11	31.26	35.38	36.68	37.84
i22	2.124	34.12	34.84	36.05	34.96	34.77
i23	2.114	29.39	28.68	26.19	38.85	38.63
i24	2.124	14.05	24.53	30.65	13.66	20
i25	2.213	26.99	26.24	35.07	35.87	38.17
i26	2.101	34.17	30.07	26.43	35.2	38.47
i27	2.114	11.76	28.66	16.67	21.89	29.7
i28	2.156	29.58	31.98	31.79	34.14	34.47
i29	2.223	22.96	25.3	34.38	32.31	32.26
i30	2.132	36.91	32.81	36.19	38.31	43.91
Average	2.15	20.59	24.35	24.75	25.48	27.96

TABLE 13. Standard deviation comparison between BMO and hybrid approaches for TSP over 30 independent runs.

Instances	BMO	BMO-HC	BMO-LAHC	BMO-SA	BMO-IG	BMO-VIG
	<i>Std</i>	<i>Std</i>	<i>Std</i>	<i>Std</i>	<i>Std</i>	<i>Std</i>
eil51	4.5	1.34	1.82	0	2.76	2.83
St70	7.1	3.2	3.9	1.1	3.89	7.6
eil76	6	3.78	3.29	1.2	4.73	8.1
Pr76	773.1	2.9	3.7	2.1	4.2	8.6
kroA100	307.7	34.8	44.58	78.91	602.14	786.22
kroB100	263	75.8	110.56	113.44	450.58	587.69
kroC100	277.5	66.36	75.57	96.83	466.56	597.1
kroD100	230.7	130.78	74.3	98.52	470.56	677.71
kroE100	282.7	98.3	78.7	114.61	288.34	78.57
rd100	127.5	41.81	37.12	45.91	154.41	221.72
eil101	6	6.16	5.27	3.92	6.72	8.69
lin105	206.6	21.56	31.9	39.95	328.04	354.49
ch130	79.6	24.72	27.19	31.06	151.5	186.25
ch150	86.8	37.31	35.95	25.8	131.56	192.4
kroA150	372.9	146.01	160.5	210.6	728.1	688.34
kroB150	335.2	122.58	110.65	120.81	584.2	806.27
D198	98.5	135.66	150.6	132.77	700.54	900.23
kroA200	383.4	205.42	201.54	151.78	1089.6	1534.3
kroB200	336.6	177.1	207.58	198.76	1270.1	1427
lin318	546.9	246.98	364.76	419.5	3459.7	3563.2
Average	236.61	79.12	86.47	94.37	544.9	631.86

the hybrid BMO-HC maintained the exploitation ability of the algorithm and continuously improve BMO until the end of the search. Parts (a), (b), (c) and (d) in Figure 2 show the ability

of the algorithm in exploring the search space followed by the ability of exploiting it (Figure 2 (e), (f), (g) and (h)). This behavior directs the algorithm toward a superior convergence.

TABLE 14. Average computational time comparison between BMO and hybrid approaches for TSP over 30 independent runs (time in second).

Instances	BMO	BMO-HC	BMO-LAHC	BMO-SA	BMO-IG	BMO-VIG
	<i>Avg. Time(s)</i>	<i>Avg. Time(s)</i>	<i>Avg. Time(s)</i>	<i>Avg. Time(s)</i>	<i>Avg. Time(s)</i>	<i>Avg. Time(s)</i>
eil51	1.17	11.46	10.54	4.08	8.85	8.73
St70	0.89	17.34	15.43	15.12	18.76	19.54
eil76	22.86	19.87	19.8	17.57	21.06	20.13
Pr76	0.21	10.12	10.76	11.21	13.76	13.22
kroA100	1	18.02	18.99	17.14	26.48	26.64
kroB100	1.58	23.76	26.83	26.56	24.31	25.76
kroC100	0.16	16.62	17.79	21.46	28.72	25.97
kroD100	1.8	23.34	20.05	25.72	29.6	32.37
kroE100	1.38	22.25	22.97	25.07	25.67	28.4
rd100	1.24	23.2	17.82	23.64	26.43	28.7
eil101	5.09	28.76	29.52	29.35	28	29.66
lin105	0.23	16.05	19.49	23.71	27.9	27.12
ch130	3	39.23	37.95	37.5	41.08	43.99
ch150	4.14	44.1	44.3	38.02	47.24	52.1
kroA150	3.49	43.54	43.61	43.21	50.67	51.93
kroB150	2.42	39.77	43.17	38.73	52.93	53.99
D198	2.33	51.17	49.45	45.19	55.14	57.33
kroA200	4.39	64.62	61.69	60.27	73.44	75.82
kroB200	4.96	61.36	63.46	59.58	71.45	75
lin318	6.51	94.95	99.72	100.7	124.22	120.48
Average	3.44	33.47	33.67	33.19	39.78	40.84

TABLE 15. P-value of Wilcoxon test of hybrid BMO algorithms against BMO for BAP.

Instance	BMO-HC vs. BMO	BMO-LAHC vs. BMO	BMO-SA vs. BMO	BMO-IG vs. BMO	BMO-VIG vs. BMO	Instance	BMO-HC vs. BMO	BMO-LAHC vs. BMO	BMO-SA vs. BMO	BMO-IG vs. BMO	BMO-VIG vs. BMO
	<i>p-value</i>	<i>p-value</i>	<i>p-value</i>	<i>p-value</i>	<i>p-value</i>		<i>p-value</i>	<i>p-value</i>	<i>p-value</i>	<i>p-value</i>	<i>p-value</i>
i01	+	+	+	+	+	i16	+	+	+	+	+
i02	+	+	+	+	+	i17	+	+	+	+	+
i03	+	+	+	+	+	i18	+	+	+	+	+
i04	+	+	+	+	+	i19	+	+	+	+	+
i05	+	+	+	+	+	i20	+	+	+	+	+
i06	+	+	+	+	+	i21	+	+	+	+	+
i07	+	+	+	+	+	i22	+	+	+	+	+
i08	+	+	+	+	+	i23	+	+	+	+	+
i09	+	+	+	+	+	i24	+	+	+	+	+
i10	+	+	+	+	+	i25	+	+	+	+	+
i11	+	+	+	+	+	i26	+	+	+	+	+
i12	+	+	+	+	+	i27	+	+	+	+	+
i13	+	+	+	+	+	i28	+	+	+	+	+
i14	+	+	+	+	+	i29	+	+	+	+	+
i15	+	+	+	+	+	i30	+	+	+	+	+

In this figure, it can be clearly seen that the algorithm tries to explore the search space until 30% of the search space and then starts to exploit it after the 30% of the search space and continues the exploitation until finding the best solution and final convergence.

To summarize, the results achieved by hybrid BMO approaches show that these approaches perform better than the BMO. In addition, the hybrid approaches matched the best-known results in the literature on 30 out of 30 instances

for BAP and on 17 out of 20 instances for TSP. From these results, it is clear that embedding single-based meta-heuristics with the BMO outperform the BMO algorithm alone. In fact, that is mainly due to the ability of single-based meta-heuristics to improve the exploitation process of the algorithm and produce good quality solutions. In addition, the statistical tests revealed that BMO-HC outperforms the other hybrid BMO approaches for BAP, whilst BMO-SA outperforms BMO-IG and BMO-VIG for TSP. However, although

TABLE 16. *P*-value of Wilcoxon test of hybrid BMO algorithms against BMO for TSP.

Instance	BMO-HC vs. DBM O	BMO-LAHC vs. BMO	BMO-SA vs. BMO	BMO-IG vs. BMO	BMO-VIG vs. BMO	Instance	BMO-HC vs. BMO	BMO-LAHC vs. BMO	BMO-SA vs. BMO	BMO-IG vs. BMO	BMO-VIG vs. BMO
	<i>p</i> -value	<i>p</i> -value	<i>p</i> -value	<i>p</i> -value	<i>p</i> -value		<i>p</i> -value	<i>p</i> -value	<i>p</i> -value	<i>p</i> -value	<i>p</i> -value
eil51	+	+	+	+	+	ch130	+	+	+	+	-
eil76	+	+	+	+	+	ch150	+	+	+	+	=
kroA100	+	+	+	+	+	kroA150	+	+	+	+	-
kroB100	+	+	+	+	+	kroB150	+	+	+	+	-
kroC100	+	+	+	+	+	kroA200	+	+	+	-	-
kroD100	+	+	+	+	+	kroB200	+	+	+	-	-
kroE100	+	+	+	+	+	lin318	+	+	+	+	-
rd100	+	+	+	+	+	rat575	+	+	+	+	+
eil101	+	+	+	+	+	rat783	+	+	+	-	-
lin105	+	+	+	+	+	fl1400	+	+	+	-	-

the results of Holm and Hochberg suggest that BMO-SA is not statistically better than BMO-HC and BMO-LAHC for TSP nevertheless, the best and the average results shown in Table 3 and Table 11 reveals that BMO-SA produced better results compared to BMO-HC and BMO-LAHC.

B. COMPARISON WITH THE STATE-OF-THE-ART APPROACHES

In the previous section, we found from analyzing the results that BMO-HC and BMO-SA outperformed other hybrid approaches in terms of solution quality for BAP and TSP, respectively, on all tested instances. In this section, we compare the performance of the of the best hybrid approaches against the most related approaches in the BAP and TSP literature. The following subsections present the comparison for BAP and TSP.

1) COMPARISON RESULTS FOR BAP

Table 8 presents the comparison results of percentage deviation according to solution quality between BMO-HC and the following approaches: Adaptive Large Neighborhood Search (ALNS) [42], Column Generation (CG) [43], Simulated Annealing with Restart Strategy (SA_{r,s}) [44], Tabu Search Algorithm (TS) [35], Iterated Greedy Heuristic (IG) [41]. Best results are highlighted in bold.

As shown in Table 8, the proposed BMO-HC performed better than 2 out of 5 algorithms and produced results same as the remaining algorithms in which they match the best-known solutions across all tested BAP instances. TS is the worse as it matched the best-known solutions on 13 out of 30 instances, while CG comes next as it matched 27 out of 30 instances. That means the BMO-HC results are very competitive and stable across all instances. This can be interpreted by the improved exploitation ability of BMO-HC which helps to search for high-performance regions of search space.

2) COMPARISON RESULTS FOR TSP

Table 9 reports the comparison results of percentage deviation according to solution quality between BMO-SA and the following approaches: discrete spider monkey optimization (DSMO) [45], swap sequence based Artificial Bee Colony algorithm (ABCSS) [46], velocity tentative PSO (VTPSO) [47], GA [17], and PSO [17].

Table 9 demonstrate that the proposed BMO-SA algorithms performed better than GA and PSO algorithms across all tested instances, and better than the remaining approaches on 12 out of 15 tested instances. Even though DSMO perform better than other approaches on 3 instances, BMO-SA still in the first place where it outperformed all other approaches on 80% of the instances. In addition, BMO-SA matched the best-known solutions on 17 out of 20 tested TSP instances whilst none of the compared approaches has matched any of best-known solutions. This is due to the impact of SA algorithm in improving the search space and attaining well-balance between exploration and exploitation of BMO.

IX. CONCLUSION

In this work, we proposed five hybrid approaches of BMO algorithm for solving combinatorial optimization problems: BMO-HC, BMO-LAHC, BMO-SA, BMO-IG, and BMO-VIG. The proposed approaches employed the single-based algorithms using low-level teamwork hybrid model.

Single-based algorithm used inside the BMO as a local search operator in which it searches the neighborhood of each solution in the population after each iteration. A hybridization probability is implemented to every solution before hybridizing to avoid early premature convergence. To evaluate the proposed methods, we tested them on the travelling salesman problem and berth allocation problem and all approaches are compared against each other. In addition, a comparative study was conducted against five recent methods in the literature including: GA, PSO, VTPSO, ABCSS, and DSMO. The computational results demonstrated that the hybrid BMO

approaches were able to outperform the original BMO across all instances in both problem domains BAP and TSP. Furthermore, the BMO-HC and BMO-SA matched the best-known solutions.

across all instances for BAP and across most of TSP instances, respectively. Also, the hybrid BMO approaches showed very competitive results when compared to the state-of-the-art methods for BAP and TSP and ranked as among the best performing approaches. In conclusion, we proved that hybridizing BMO with single-based algorithms managed to improve its exploitation and attain the well-balance between exploitation and exploration of the algorithm.

In future work, it might be beneficial to investigate the possibility of proposing an online framework of the hybrid BMO to solve different combinatorial optimization problem without the need to select the single-based algorithm manually. On other words, proposing a BMO model that can automatically adjust the components of the hybrid BMO according to the search status.

APPENDIX A

See Tables 10–16.

REFERENCES

- [1] H. Masum, "Review of how to solve it: Modern heuristics Zbigniew Michalewicz and David B Fogel," *ACM SIGACT News*, vol. 32, no. 1, 2001.
- [2] F. Hutter, H. H. Hoos, L. B. Kevin, and T. Stützle, "ParamILS: An automatic algorithm configuration framework," *J. Artif. Intell. Res.*, vol. 36, no. 1, pp. 267–306, 2009, doi: [10.1613/jair.2861](https://doi.org/10.1613/jair.2861).
- [3] T. Uri, "Hybrid meta-heuristics for combinatorial optimization," *Constraints*, vol. 20, p. 473, 2015, doi: [10.1007/s10601-015-9209-7](https://doi.org/10.1007/s10601-015-9209-7).
- [4] M. B. Dowlatshahi, H. N. Pour, and M. Mashinchi, "A discrete gravitational search algorithm for solving combinatorial optimization problems," *Inf. Sci.*, vol. 258, no. 2, pp. 94–107, Feb. 2014, doi: [10.1016/j.ins.2013.09.034](https://doi.org/10.1016/j.ins.2013.09.034).
- [5] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1983, doi: [10.1126/science.220.4598.671](https://doi.org/10.1126/science.220.4598.671).
- [6] E. Talbi, F. Yalaoui, and L. Amode, *Metaheuristics for Production Systems*, vol. 60. New York, NY, USA: Springer, 2016.
- [7] E. K. Burke and Y. Bykov, "The late acceptance Hill-Climbing heuristic," *Comput. Sci. Math., Univ. Stirling, Stirling, U.K., Tech. Rep. CSM-192*, 2017.
- [8] M. K. Dhadwal, S. N. Jung, and C. J. Kim, "Advanced particle swarm assisted genetic algorithm for constrained optimization problems," *Comput. Optim. Appl.*, vol. 58, no. 3, pp. 781–806, Jul. 2014, doi: [10.1007/s10589-014-9637-0](https://doi.org/10.1007/s10589-014-9637-0).
- [9] E. T. Yassen, M. Ayob, M. Z. A. Nazri, and N. R. Sabar, "Meta-harmony search algorithm for the vehicle routing problem with time windows," *Inf. Sci.*, vol. 325, pp. 140–158, Dec. 2015, doi: [10.1016/j.ins.2015.07.009](https://doi.org/10.1016/j.ins.2015.07.009).
- [10] R. Ruiz and T. Stützle, "A simple and effective iterated greedy algorithm for the permutation flowshop scheduling problem," *Eur. J. Oper. Res.*, vol. 177, no. 3, pp. 2033–2049, Mar. 2007, doi: [10.1016/j.ejor.2005.12.009](https://doi.org/10.1016/j.ejor.2005.12.009).
- [11] L. W. Jacobs and M. J. Brusco, "Note: A local-search heuristic for large set-covering problems," *Nav. Res. Logistics*, vol. 42, no. 7, pp. 1129–1140, 1995.
- [12] E. Marchiori and A. Steenbeek, "An evolutionary algorithm for large scale set covering problems with application to airline crew scheduling," in *Workshops on Real-World Applications of Evolutionary Computation*. Berlin, Germany: Springer, 2000, doi: [10.1007/3-540-45561-2_36](https://doi.org/10.1007/3-540-45561-2_36).
- [13] M. Dorigo and L. M. Gambardella, "Ant colonies for the travelling salesman problem," *Biosystems*, vol. 43, no. 2, pp. 73–81, 1997, doi: [10.1016/S0303-2647\(97\)01708-5](https://doi.org/10.1016/S0303-2647(97)01708-5).
- [14] C. Blum, M. Dorigo, and T. Stützle, *Ant Colony Optimization*. Cambridge, MA, USA: MIT Press, 2004, p. 300.
- [15] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," *Proc. 6th Int. Symp. Micro Mach. Hum. Sci.*, 1995, pp. 39–43, doi: [10.1109/MHS.1995.494215](https://doi.org/10.1109/MHS.1995.494215).
- [16] A. Askarzadeh, "Bird mating optimizer: An optimization algorithm inspired by bird mating strategies," *Commun. Nonlinear Sci. Numer. Simul.*, vol. 19, no. 4, pp. 1213–1228, 2014, doi: [10.1016/j.cnsns.2013.08.027](https://doi.org/10.1016/j.cnsns.2013.08.027).
- [17] A. Arram, M. Ayob, G. Kendall, and A. Sulaiman, "Bird mating optimizer for combinatorial optimization problems," *IEEE Access*, vol. 8, pp. 96845–96858, 2020.
- [18] Q. Zhang, G. Yu, and H. Song, "A hybrid bird mating optimizer algorithm with teaching-learning-based optimization for global numerical optimization," *Statist., Optim. Inf. Comput.*, vol. 3, no. 1, pp. 54–65, Feb. 2015, doi: [10.19139/soic.v3i1.86](https://doi.org/10.19139/soic.v3i1.86).
- [19] T.-C. Ou, W.-F. Su, X.-Z. Liu, S.-J. Huang, and T.-Y. Tai, "A modified bird-mating optimization with hill-climbing for connection decisions of transformers," *Energies*, vol. 9, no. 9, p. 671, Aug. 2016, doi: [10.3390/en9090671](https://doi.org/10.3390/en9090671).
- [20] A. Arram, M. Z. A. Nazri, M. Ayob, and A. Abunadi, "Bird mating optimizer for discrete berth allocation problem," in *Proc. Int. Conf. Electr. Eng. Informat. (ICEEI)*, Aug. 2015, pp. 450–455, doi: [10.1109/ICEEI.2015.7352543](https://doi.org/10.1109/ICEEI.2015.7352543).
- [21] K. Bryant and A. Benjamin, "Genetic algorithms and the traveling salesman problem by kylie bryant genetic algorithms and the traveling salesman problem," in *Proc. 1st GNT Reg. Conf. Math. Stat. Appl.*, Dec. 2000.
- [22] C.-K. Ting, C.-H. Su, and C.-N. Lee, "Multi-parent extension of partially mapped crossover for combinatorial optimization problems," *Expert Syst. Appl.*, vol. 37, no. 3, pp. 1879–1886, Mar. 2010, doi: [10.1016/j.eswa.2009.07.082](https://doi.org/10.1016/j.eswa.2009.07.082).
- [23] E. G. Mahadevan, *Ammonium Nitrate Explosives for Civil Applications: Slurries, Emulsions and Ammonium Nitrate Fuel Oils*, vol. 74. Hoboken, NJ, USA: Wiley, 2009.
- [24] M. Lones, "Sean luke: Essentials of metaheuristics," *Genet. Program. Evolvable Mach.*, vol. 12, no. 3, pp. 333–334, Sep. 2011.
- [25] E. K. Burke and Y. Bykov, "A late acceptance strategy in hill-climbing for exam timetabling problems," in *Proc. PATAT Conf.*, Montreal, QC, Canada, 2008, pp. 1–7.
- [26] V. Černý, "Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm," *J. Optim. Theory Appl.*, vol. 45, no. 1, pp. 41–51, Jan. 1985, doi: [10.1007/BF00940812](https://doi.org/10.1007/BF00940812).
- [27] C.-C. Lo and C.-C. Hsu, "An annealing framework with learning memory," *IEEE Trans. Syst., Man, Cybern. A, Syst. Humans*, vol. 28, no. 5, pp. 648–661, Sep. 1998, doi: [10.1109/3468.709611](https://doi.org/10.1109/3468.709611).
- [28] J. M. Framinan and R. Leisten, "Total tardiness minimization in permutation flow shops: A simple approach based on a variable greedy algorithm," *Int. J. Prod. Res.*, vol. 46, no. 22, pp. 6479–6498, Nov. 2008, doi: [10.1080/00207540701418960](https://doi.org/10.1080/00207540701418960).
- [29] P. Hansen, N. Mladenović, J. Brimberg, and J. A. M. Pérez, "Variable neighborhood search," *Int. Ser. Oper. Res. Manag. Sci.*, vol. 272, no. 11, pp. 57–97, 2019, doi: [10.1007/978-3-319-91086-4_3](https://doi.org/10.1007/978-3-319-91086-4_3).
- [30] Y. Wang, "The hybrid genetic algorithm with two local optimization strategies for traveling salesman problem," *Comput. Ind. Eng.*, vol. 70, pp. 124–133, Apr. 2014, doi: [10.1016/j.cie.2014.01.015](https://doi.org/10.1016/j.cie.2014.01.015).
- [31] G. Laporte, "The traveling salesman problem: An overview of exact and approximate algorithms," *Eur. J. Oper. Res.*, vol. 59, no. 2, pp. 231–247, Jun. 1992, doi: [10.1016/0377-2217\(92\)90138-Y](https://doi.org/10.1016/0377-2217(92)90138-Y).
- [32] A. Arram, M. Ayob, and M. Zakree, "Comparative study of meta-heuristic approaches for solving traveling salesman problems," *Asian J. Appl. Sci.*, vol. 7, no. 7, pp. 662–670, Sep. 2014.
- [33] G. Reinelt, *The Traveling Salesman: Computational Solutions for TSP Applications*. New York, NY, USA: Springer-Verlag, 1994.
- [34] C. Bierwirth and F. Meisel, "A follow-up survey of berth allocation and quay crane scheduling problems in container terminals," *Eur. J. Oper. Res.*, vol. 244, no. 3, pp. 675–689, 2015, doi: [10.1016/j.ejor.2014.12.030](https://doi.org/10.1016/j.ejor.2014.12.030).
- [35] J.-F. Cordeau, G. Laporte, P. Legato, and L. Moccia, "Models and tabu search heuristics for the berth-allocation problem," *Transp. Sci.*, vol. 39, no. 4, pp. 526–538, Nov. 2005, doi: [10.1287/trsc.1050.0120](https://doi.org/10.1287/trsc.1050.0120).
- [36] C.-J. Ting, K.-C. Wu, and H. Chou, "Particle swarm optimization algorithm for the berth allocation problem," *Expert Syst. Appl.*, vol. 41, no. 4, pp. 1543–1550, 2014, doi: [10.1016/j.eswa.2013.08.051](https://doi.org/10.1016/j.eswa.2013.08.051).

- [37] W. L. Zhong, J. Zhang, and W. N. Chen, "A novel discrete particle swarm optimization to solve traveling salesman problem," in *Proc. IEEE Congr. Evol. Comput.*, Sep. 2007, pp. 3283–3287, doi: [10.1109/CEC.2007.4424894](https://doi.org/10.1109/CEC.2007.4424894).
- [38] N. R. Sabar and G. Kendall, "Aircraft landing problem using hybrid differential evolution and simple descent algorithm," in *Proc. IEEE Congr. Evol. Comput.*, Jul. 2014, pp. 520–527, doi: [10.1109/CEC.2014.6900251](https://doi.org/10.1109/CEC.2014.6900251).
- [39] E. Soubeiga, "Development and application of hyperheuristics to personnel scheduling," Ph.D. dissertation, School Comput. Sci., Univ. Nottingham, Nottingham, U.K., Jun. 2003.
- [40] D. Ouelhadj and S. Petrovic, "A cooperative hyper-heuristic search framework," *J. Heuristics*, vol. 16, no. 6, pp. 835–857, Dec. 2010, doi: [10.1007/s10732-009-9122-6](https://doi.org/10.1007/s10732-009-9122-6).
- [41] S.-W. Lin, K.-C. Ying, and S.-Y. Wan, "Minimizing the total service time of discrete dynamic berth allocation problem by an iterated greedy heuristic," *Sci. World J.*, vol. 2014, pp. 1–12, Oct. 2014, doi: [10.1155/2014/218925](https://doi.org/10.1155/2014/218925).
- [42] G. R. Mauri, G. M. Ribeiro, L. Antonio, N. Lorena, and G. Laporte, "An adaptive large neighborhood search for the discrete and continuous berth allocation problem," *Comput. Oper. Res.*, vol. 70, pp. 140–154, Jun. 2016, doi: [10.1016/j.cor.2016.01.002](https://doi.org/10.1016/j.cor.2016.01.002).
- [43] G. R. Mauri, A. C. M. Oliveira, and L. A. N. Lorena, "A hybrid column generation approach for the berth allocation problem," in *Proc. Eur. Conf. Evol. Comput. Combinat. Optim.* Berlin, Germany: Springer, 2008.
- [44] S.-W. Lin and C.-J. Ting, "Solving the dynamic berth allocation problem by simulated annealing," *Eng. Optim.*, vol. 46, no. 3, pp. 308–327, Mar. 2014, doi: [10.1080/0305215X.2013.768241](https://doi.org/10.1080/0305215X.2013.768241).
- [45] M. A. H. Akhand, S. I. Ayon, S. A. Shahriyar, N. Siddique, and H. Adeli, "Discrete spider monkey optimization for travelling salesman problem," *Appl. Soft Comput.*, vol. 86, Jan. 2020, Art. no. 105887, doi: [10.1016/j.asoc.2019.105887](https://doi.org/10.1016/j.asoc.2019.105887).
- [46] I. Khan and M. K. Maiti, "A swap sequence based artificial bee colony algorithm for traveling salesman problem," *Swarm Evol. Comput.*, vol. 44, pp. 428–438, Feb. 2019, doi: [10.1016/j.swevo.2018.05.006](https://doi.org/10.1016/j.swevo.2018.05.006).
- [47] M. A. H. Akhand and M. A. Rashid, "Velocity tentative particle swarm optimization to solve TSP," in *Proc. Int. Conf. Elect. Inf. Commun. Technol. (EICT)*, Feb. 2014, pp. 1–6, doi: [10.1109/EICT.2014.6777868](https://doi.org/10.1109/EICT.2014.6777868).



ANAS ARRAM received the Ph.D. degree in computer science from the National University of Malaysia in 2017. He is a member of Data Mining and Optimization Research Group (DMO), National University of Malaysia. He is also an assistant professor at An-Najah National University, Palestine. His main research areas include meta-heuristics, hyper-heuristics, scheduling and timetabling, machine learning, and deep learning.



MASRI AYOB received the Ph.D. degree in computer science from the University of Nottingham, in 2005. She was a member of the ASAP Research Group, University of Nottingham. She has been a Lecturer with the Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia (UKM), since 1997, where she is currently a Principle Researcher with the Centre for Artificial Intelligent (CAIT), Data Mining and Optimisation Research Group (DMO). She has published more than 150 papers at international journals and peer-reviewed international conferences. Her main research interests include meta-heuristics, hyper-heuristics, and scheduling and timetabling, especially educational timetabling, healthcare scheduling, and routing problems. She has been a program committee member for more than 50 international conferences and a reviewer for high-impact journals.



ALAA SULAIMAN received the B.A. degree in electrical and computer engineering from An-Najah National University, Nablus, Palestine, in 2003, and the M.Sc. degree in scientific computing from Birzeit University, Ramallah, Palestine, in 2010, with a focus on computer science and engineering. He is currently pursuing the Ph.D. degree with the Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia (UKM). His area of interests include machine learning, image processing, and pattern recognition.

...