DOI: 10.6977/IJoSI.202109_6(5).0004
A. M. Jabbar, K R. Ku-Mahamud/ Int. J. Systematic Innovation, 6(5), 32-43 (2021)

Hybrid Black Widow Optimization and Variable Neighborhood Descent Algorithm for Traveling Salesman Problem

Ayad Mohammed Jabbar[1]*, Ku Ruhana Ku-Mahamud[2]
[1] Computer Science Department, Shatt Al-Arab University College, Iraq
[2] School of Computing, Universiti Utara Malaysia, Malaysia
* Corresponding author E-mail: ayadjb@gmail.com
(Received 11 July 2021; Final version received 22 July 2021; Accepted 7 September 2021)

Abstract

Local search algorithms in general are better than population-based algorithms in the terms of exploitation capability in finding more local regions in the search space which provide more ability to explore search space in finding global regions. Black widow optimization (BWO) algorithm is one of the best population-based algorithms which was proposed in 2020 to solve engineering optimization problems. However, this algorithm has a limitation in the exploitation of search space and reactivate a search when stagnation occurs during the algorithm run. Thus, deep search and effectively exploring the search space are not possible during the algorithm run. To overcome these drawbacks, this study proposes two modifications to the BWO algorithm. The first modification is the integration of variable neighborhood descent used to enhance the exploitation process in finding more local regions in the neighborhood during the algorithm run. The second modification focuses on the reactive search process by integrating a new convergence indicator for the algorithm during the algorithm run and online reactive search process. Two benchmark datasets were used to evaluate the proposed modification. The minimum tour distance provided by each algorithm has been used as the performance metric in determining the credibility of the hybrid BWO algorithm and results have been compared with best-known algorithms include African buffalo optimization (ABO), ant colony optimization (ACO), artificial bee colony (ABC), particle swarm optimization (PSO) and a hybrid algorithm consisting of harmony search, particle swarm and ACO (HPSACO). The hybrid BWO algorithm has produced better minimum tour distance compared to ABO, ACO, ABC, PSO and HPSACO algorithms which demonstrate that the hybrid BWO can be applied to solve several optimization problems including vehicle routing problem, classification and clustering.

Keywords: Exploration, Exploitation, Local search, Neighborhood search, Swarm algorithms, Traveling salesman problem.

1. **Introduction**

The optimal solution in the field of artificial intelligence refers to the best solution that can be obtained from the search space compared with other several solutions provided from the same search space (Desale et al., 2015; Jabbar, Ku-Mahamud, & Sagban 2019a, 2019c). This type of solution can be found when solving any NP-hard problems that have complex search space have several landscapes. Examples of NP-hard problems are classification (Stegherr, Heider, & Hähner, 2020), clustering (Hossain et al., 2019), and feature selection (Venkatesh & Anuradha, 2019). The search space of the problem is difficult to solve within the estimated time and requires special algorithm to provide a stochastic search guided by the objective function and randomness covering a wide area of search space (Dao, Abhary, & Marian, 2015). This kind of search is the main foundation of algorithms and known as metaheuristics, which combines heuristic methods in high-level metaphors to find optimal or near-optimal solution in a reasonable time. Examples of metaphors in real life include foraging behavior, memory, annealing, evolution, reproduction style and cannibalism, these metaphors can be represented by several algorithms including ACO (Al-Behadili, Sagban, & Ku-Mahamud, 2020a), tabu search (TS) (Ghany et al., 2020), simulated annealing (SA) (Moriguchi, Ueki, & Saito, 2015), genetic algorithm (GA) (Das & Pratihar, 2018), and black widow optimization algorithm (BWO) (Hayyolalam & Pourhaji Kazem, 2020; Abuhamdah, 2020). The BWO is one of the swarm intelligence algorithms inspired by black widow spiders, which simulates the spider mating process in nature (Houssein et al., 2020). The algorithm has three important stages which simulates the real behavior of black widow. These stages include mating, which starts when the male enters the web of female; reproduction and cannibalism, which start by hatching the egg and offspring engagement; and increasing the density of the population by keeping only strong spiders which is known as sibling cannibalism. However, the main issue is how to achieve better exploitation and exploration of the search space and avoiding local optima problem in the process to obtain the optimal result rapidly and with high accuracy. The BWO the algorithm has limitations in terms of exploitation of the search space and reactivating a search when stagnation occurs during the algorithm run. The algorithm should improve the population iteratively until it converges to local optima and then the reactive the search process by updating its population with new fresh population located far from the region that has that local optimum. Considering this limitation, scholars explore other algorithms to solve the limitation which resulted in a hybrid algorithm. This process can be achieved using metaheuristics algorithms, which use the neighborhood change. An example of these algorithms is the variable neighborhood descent (VND), which descent to the regions that have local optima and can escape from these regions according to the designed VND framework (Duarte et al., 2016). In the optimization problem, the region that has good solution certainly contains neighborhood regions that have better solutions. Thus, several close neighborhoods should be explored to find global solutions by generating several landscapes using the VND algorithm. The algorithm has a befit of generating several landscapes during the algorithm run whereby it can increase the diversity of solutions with different solutions. However, an important issue that should be considered is moving from one landscape to another. Despite of its importance, it is not always sufficient when no knowledge about the exploration state is provided. Hence, more time is required to perform exploitation than exploration. Based on this consideration, two modifications are proposed. VND is used to enhance the performance of the BWO algorithm in terms of exploitation capability to find more solutions around the best regions in the search space. VND enhances the BWO algorithm to find more local regions by improving the neighborhood search during the algorithm run. Reactive search process is proposed as the second modification by integrating a new indicator of the convergence of the algorithm during the algorithm run. The reactive search will enhance the search process by moving the search into new promising region and keeping the history of the search to use them as a guide for future search in advanced iterations. Both modifications will enhance the balance between the exploration and exaptation. Finally, the performance of the proposed algorithm will increase. These two modifications are recruited in BWO to avoid convergence because of its limitation in exploiting the search in finding more local regions at the best so far region and reactivating the search space during the algorithm run. Reactive search is integrated into the algorithm, which will automatically reactivate the search process when algorithm falls into local optima or converges to the same solution.

This article is organized as follows. Section 2 shows the related works of swarm based-algorithms. Section 3 elaborates the proposed hybrid BWO algorithm with its formulation for the traveling salesman's problem. Section 4 describes the benchmark datasets used in the experimental results. Section 5 states the conclusion and future works.

2. Related works

Solving NP hard problem such as TSP where finding the minimum tour distance between all cities is not an easy task, especially for a large number of cities and the search space is complex. Finding an optimal solution where the objective function reaches its minimum value at an acceptable time has several landscapes. This process of reaching the optimal solution is sometimes useless, as the time required to solve these problems may exceed the usefulness of the solution. Here the need arose to use algorithms that have the ability to produce good solutions in an ideal time. Those algorithms are categorized as optimization approaches targeted at finding near or optimal solutions in an ideal time. The optimization approach shown in Fig. 1, classified into several approaches which include estimation approach, exact approach, and approximate approach. The exact approach requires an exponential time to solve the hard problem due to its process to find all solutions. The estimation approach uses a previously defined range of inputs such as used in parameter problem to solve the problem according to the defined inputs. The last approach is the approximate approach can be classified into single-based solution such as local search and population-based techniques (Al-Behadili, Ku-Mahamud, & Sagban, 2020b; Duarte et al., 2016). Local search methods, such as TS and SA, perform neighborhood search to modify single solution by exchanging segments of its components to produce better solutions. Meanwhile, in the population-based techniques such as ACO and BWO, more than one solution are used iteratively during the algorithm run (Al-Behadili, Ku-Mahamud, & Sagban, 2019c; Sicilia et al., 2016). The search process is guided in different processes according to metaphor characteristics employed in the algorithm. Examples of these characteristics' areas are the acceptance criterion and cooling schedule in SA algorithm (Yang & Yang, 2014), neighbor choice in TS algorithm (Zhou et al., 2013), recombination, mutation, and selection in GA algorithm (AL-Behadili, Ku-Mahamud, & Sagban, 2020a; Stegherr et al., 2020), mating, reproduction and cannibalism (Rasekh & Osawa, 2020), siblings in BWO, and pheromone update and probabilistic construction in ACO algorithm (Jabbar, Ku-Mahamud, & Sagban, 2019b). However, there is a big deference between both approaches regarding the stigmergy which is a key role in the nurturing of society that does not exist in the evolutionary-based approach. The swarm-based approach includes different algorithm such as ACO, artificial bee colony (ABC), particle swarm optimization (PSO) and BWO. These algorithms have the stigmergy principle that represents the medium for information transformation. A typical example is pheromone trails that leads to organization of ants in ACO, while in BWO represents the attraction between the male and female. However, the main differences between each algorithm are the algorithm memory and how to represent the search space of the problem. An example of that PSO its memory represents the population of particles, ACO is represented by pheromone matrix that retain the information of ants and ABC represented by its population of bees. The BWO algorithm followed the same procedure of the swarm algorithms, where its memory is represented by the population of the black widows.

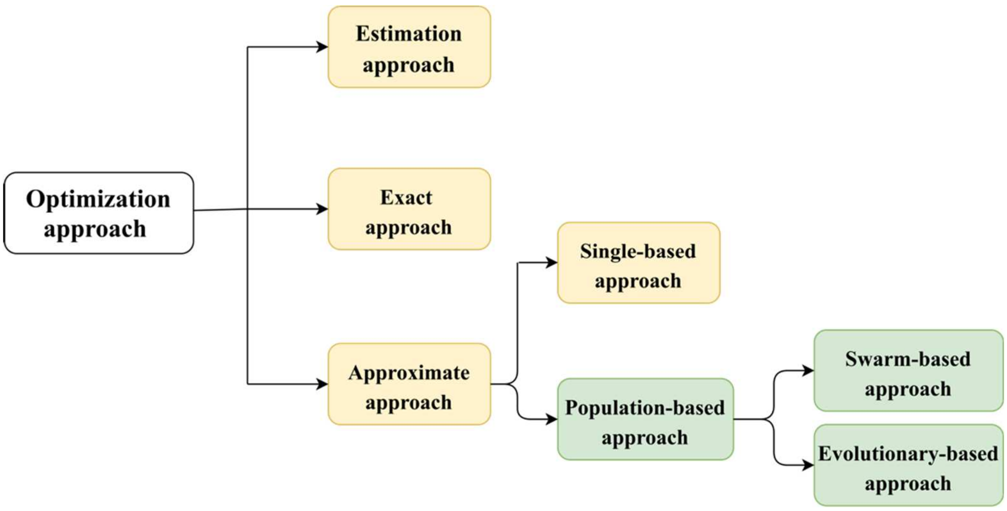

Fig. 1. Taxonomy of optimization approach

Several related works have been proposed in the literature such as Tan et al. (2020) who proposed a hybrid POS algorithm and hill climbing (HC) for high school timetabling problem. The proposed algorithm has two modifications in PSO and HC algorithms. The first modification is the solution transformation of the mutation and crossover operations while the second modification is to increase the efficiency of the exploration and exploitation in search space using HC algorithm. However, it is found that the HC algorithm only accept the candidate solutions that have better fitness. Thus, it limits the exploration capability of PSO algorithm in the terms of looking for global candidate solutions. Other similar research was proposed by Goh et al. (2020) as a hybrid local search algorithm to address the post enrolment course timetabling. There are two phases in the proposed hybrid local search algorithm. The first phase is to find a feasible solution, while the second phase focuses on minimizing the soft constraint of the generated feasible solution from the first phase. In finding a feasible solution, the tabu search (TS) with sampling hybrid algorithm and perturbation with iterated local search (ILS) hybrid algorithm were employed. Simulated annealing with reheating (SAR) algorithm and two preliminary runs (SAR-2P) algorithm are proposed to minimize the soft constraint of the feasible solution. The proposed algorithm has a drawback in terms of exploration capability, but in other aspects it shows promising results. Thus, it requires other algorithm such as ACO algorithm to overcome the exploration problem. Another related research has proposed a re-randomization method coupled with variable neighborhood search (VNS) to solve the optimal allocation of a fixed set of experimental units (Hore, Dewanji, & Chatterjee, 2016). The re-randomization method increases the probability to find more reasonable initial allocations by incorporating the randomization during the search process. However, it would be better to include various criteria to control the randomization of the search process and to avoid losing the exploitation part. An algorithm by coupling variable neighborhood search (VNS) with stochastic search to improve the exploration of VNS has been proposed by Hore, Chatterjee, and Dewanji (2018). The purpose of this coupling is to avoid the problem of local optima solution provided by VNS. Three modifications have been added to increase the stochastic search of the algorithm. These are the initial tour, construction of neighborhood and new stopping criteria. Although the proposed algorithm showed promising result, its time complex was long. Other related research to improve VNS by keeping the characteristics of the best solution during the algorithm run which often be kept and used to obtain promising neighboring solutions (Hore, Dewanji, & Chatterjee, 2014). This kind of VNS algorithm has recently been successfully applied in the field of design of experiments by adding optimum allocation of experimental units with known predictors into two treatment groups. However, adding local search to swarm algorithms is required because of the exploitation capability in finding more local regions. From the literature, swarm algorithms such as BWO remains poor as its exploitation strategy is incapable to intensify the search of local regions. Thus, integration of a local search algorithm such as VND is required with an indicator to check the convergence state of the algorithm. This indicator reactivates the search process during the algorithm run and provide more optimal solutions.

3. Proposed hybrid algorithm

Black widow optimization algorithm, which was proposed by Hayyolalam and Pourhaji Kazem (2020) is one of the best population-based swarm algorithms to solve NP-hard combinatorial problems (Sathish & Ananthapadmanabha, 2021; Sheriba & Rajesh, 2021). The algorithm is inspired by the black widow spider's nature behavior representing the mating process in spiders. The algorithm has three important stages which simulates the real behavior of the black widow. These stages are classified as follows the matting process of black widow spider, the reproduction style and cannibalism and the sibling cannibalism.

The mating process starts when the male enters the web of the female. This process occurs when a female black widow desire to mate with one of the males by attracting them using her pheromone. This process is followed by laying her eggs on the sock and wait for hatching. The following stage is reproduction style and cannibalism, which start by egg hatching and offspring engagement. Immediately during or after the matting, the male is consumed by the female black widow, which is a natural behavior that can be seen in different invertebrate societies. Researchers believe that the behavior of male might confer the chance to increase the number of eggs, resulting in greater chance of continuity of offspring. The next stage is where eggs are hatched and spend the time on the web feeding on the yolk and molt. During this time, an important natural behavior known as sibling cannibalism can be observed. This stage increases the density of the population by keeping only strong spiders. The strong spider eats the weak siblings, female black widow eats her husband, and other case such as other spiders consume their mothers. All kind of sibling cannibalism changes the diversity of the population, thus new generation can be better than the older generation. The BWO algorithm employs sibling cannibalism to improve the search process during the run by achieving only high-quality solutions. However, not all cases are desirable in optimization problems, such as improving the quality of solution guided by the objective function will lose its diversification capability (Stützle & Hoos, 2000). In the same manner, high diversification forces the algorithm to lose its exploitation capability. This research proposes a modification which includes a reactive search space during the algorithm run and simultaneously finds the optimal local regions. The mechanism is established by copying the best individuals from the original population and keep them in a temporary memory. The algorithm starts to improve existing individuals in temporary memory until no further improvement can be obtained. This step is considered as the exploration step when reactive search is performed, moving the search process to another promising global region. In each step of the improvement, the best individuals move from the original population to the temporary memory. Through this step, the second modification is added, which apply VND to increase the process of finding more promising local regions in the search space. Variable neighborhood descent enhances the BWO algorithm to find more local regions by improving the neighborhood search during the algorithm run through generation of more landscapes. It moves the best solutions generated during the run and checks when stagnations occur to restart the history of search process as shown in Fig. 2.

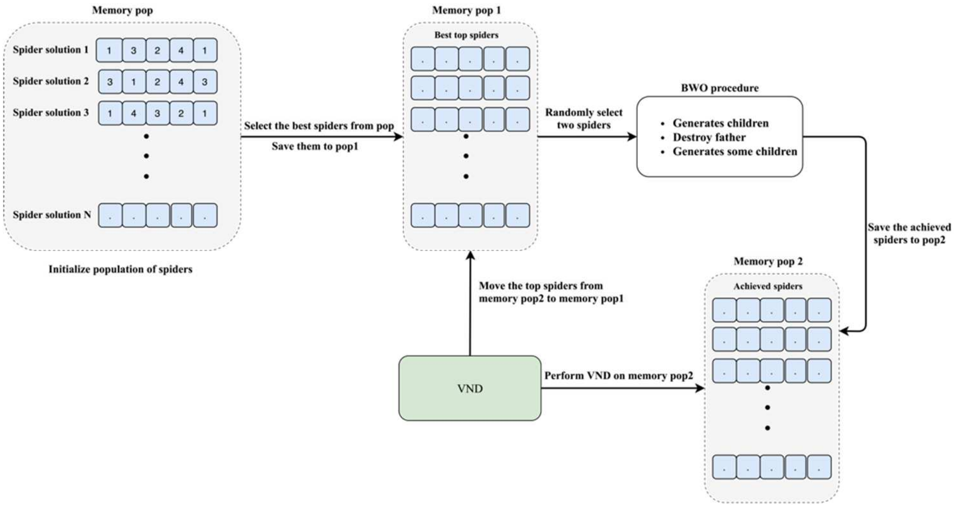

Fig. 2. The mechanism of the modifications

The algorithm starts by initializing the population of spiders in the memory pop (single spider represents single solution). The memory pop1 aims to keep the best solutions during the algorithm run and use them to refresh the search space. This is established after copying the best solutions (best spiders) from the population pop and save them to pop1. The process of the algorithm starts by selecting two solutions, then generate the children and destroy the fathers and some children. The rest of the solutions after this process is kept in the memory called pop2. The next process performs VND to find more optimal solutions and the best produced solution is select from pop2 and saved in pop1. This process ensures deep searching in the local regions of the best solutions, thereby finding the best neighborhood of all in pop2. Once the neighborhood is determined, the best top solutions in pop2 are moved to pop1. However, if convergence occurs in the algorithm, then it will deselect the best solution from pop and save them in pop1, thereby reactivating the search process during the algorithm run. Variable neighborhood descent is used in this research due to its capability of convergence to the optimal solution during long time, giving more time to explore the search space. Different landscapes refer to different neighborhood structures, thus different solutions can be used to avoid the algorithm being trapped at local optima problem. In this research, two operations are performed to provide several landscapes including the pair-swap and inversion. The pair-swap will swap two cities in randomly manner, while the inversion operation inverts a subsequence of cities between two swapped cities which are randomly selected from the solution. This algorithm has the benefit of generating several landscapes during the algorithm run, and it can increase the probability of diversity of solutions and avoid the case of losing the diversity in the population and other memories. VND generates several landscapes iteratively. Thus, the algorithm can use different neighborhood during the run to avoid the local optima as shown in Fig. 3 (El-Ghazali Talb, 2009). The VND algorithm starts by generating a set of different neighborhood structures Nl (l=1,…,lmax). Let N1 be the first neighborhood located in the local regain of the initial solution x. Thus, if no improvement in the current local region based on the fitness function of N1 compared with x occurs, then the neighborhood structure will be changed from N1 to N1+1 until the best neighborhood structure is found. Otherwise, the fitness function of x is kept. The pseudo-code of the variable neighborhood decent algorithm is illustrated in Fig. 4 (El-Ghazali Talb, 2009).

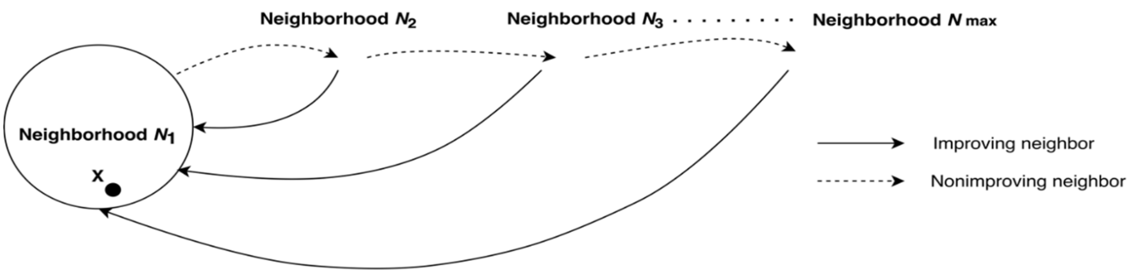

Fig. 3. Principle of variable neighborhood descent algorithm

VND algorithm	
Input	A set of neighborhood structures $N_l(l = 1, \ldots, l_{max})$.
Output	The best found solution $(best)$
Step :1	$x = x_0$ %% generate the initial solution
Step :2	**repeat**
Step :3	$l = 1$
Step :4	**while** $l \le l_{max}$ **do**
Step :5	Find the best neighbor $x^{`}$ of x in $N_{l(x)}$
Step :6	**if** $f(x^{`}) < f(x)$ **then**
Step :7	$x = x^{`}$
Step :8	$l = 1$
Step :9	**else**
Step :10	$l = l + 1$
Step :11	**end-if**
Step :12	**end-while**
Step :13	**until** no improvement exists in any l_{max} neighborhoods
Step :14	**return** x;

Fig. 4. Pseudo-code of VND algorithm

VND selects a single solution x from the memory in sequence for further improvement by generating a set of neighborhoods and select the first obtained neighborhood that increases the quality of solution (better minimum tour distance). The output of the VND algorithm is a solution has better quality replaced with the original solution x. This process is iteratively performed, as shown in step 6 of Fig. 4 until the best-found solution is obtained in step 14. The complete process of the proposed hybrid BWO algorithm is illustrated in Fig.

Proposed hybrid BWO algorithm	
Input	TSP dataset
Output	A minimum tour distance s
Step :1	Initialize all parameters (pr, nr, nm, q_0, p، $limit$, N_l)
Step :2	Initialize population of spiders pop and evaluate each one
Step :3	Select best nr solutions in pop and save them in $pop1$ based on procreating rate (pr)
Step :4	**while** ($iteration$ $<$ N_l) **do**
Step :5	**repeat**
Step :6	Randomly select two solutions as parents from $pop1$
Step :7	Generates D children
Step :8	Destroy father
Step :9	Based on the cannibalism rate (CR), destroy some children
Step :10	Save the remain solutions into $pop2$
Step :11	**until** stopping nv is met
Step :12	Perform VND on $pop2$ and copy the best nr solutions in $pop2$ and save them in $pop1$
Step :13	**if** (av(solutions in old $pop2$) == av(solutions in current $pop2$)) **then**
Step :14	$limit++$
Step :15	**end-if**
Step :16	**if** ($limit >$ $limit$ max) **then**
Step :17	Select best nr solutions in pop and save them in $pop1$
Step :18	**end-if**
Step :19	**repeat**
Step :20	Select a solution from $pop1$
Step :21	Mutate randomly one chromosome of the solution and generate a new solution
Step :22	Save the new one into $pop3$
Step :23	**until** stopping nm is met
Step :24	Update pop=$pop2$+$pop3$
Step :25	**end-while**
Step :26	**return** s

Fig. 5. Pseudo-code of proposed hybrid BWO algorithm

The hybrid BWO starts by pop1 initializing all parameters and the population of spiders, pop, in steps 1 and 2. In step 3, the algorithm selects the best nr solutions from pop and save them to, which represents the best solutions in the population to be matted latter. In step 4, the algorithm starts its iterations compared with the maximum number of iterations N_l. In step 5, the algorithm starts its cycling compared with its procreation rating to the maximum allowed procreation which is equals to 50 in this research. The next step is to randomly select two solutions from pop1. In the current time, the best nr solutions exist in pop1. Steps 7 and 8 include matting using the crossover operator between two individuals and destroying the father after the matting process. To introduce more diversity, several children and mothers are destroyed in step 9 and save in pop2 as shown in step 10. The next step highlights the proposed modification where VND is used on all solutions in pop2. However, in general, all local search algorithms provide good solution but is easily stacked at the local optima. Thus, in this research, another modification is added to check if the algorithm has sufficient diversity solution or not and to decide reactivating the search space by adding new solutions into pop1 from pop, as shown in steps 12-18. In the following steps, the algorithm employs mutation to improve the solution. In step 24, the algorithm will update pop by using all solutions that exist in pop2 and pop3. The output of the algorithm is the best solution found during the run as shown in in step 26.

4. Experiment and evaluation

Prepare The experiments have been conducted in two scenarios to evaluate the performance of the hybrid BWO algorithm. In each scenario, different set of TSP datasets (Jia, 2015; Odili & Mohmad Kahar, 2016) were used. The first set includes eight datasets (berlin52, st70, eil76, pr76, kroa100, eil101, ch150, and tsp225). These datasets have been used in a study conducted by Odili and Mohmad Kahar (2016) and the results also available in (TSPLIB, 1995). Results from the experiments conducted on the first set of the datasets have been compared to African buffalo optimization (ABO), ACO, and artificial bee colony (ABC). The second set consists of ten (10) benchmark datasets including the att48, st70, eil76, pr152, gil262, rd400, pr1002, d1291, fnl4461, and brd14051 (Jia, 2015) and results are available in TSPLIB (1995). The algorithms that have been used in the second part of the experiment are ABO, ACO, PSO, and harmony search, particle swarm and ACO (HPSACO) algorithms. Table 1 shows 16 TSP datasets that differ in the number of cities and the optimal distance for each dataset. Note that both st70 and eil70 are mentioned in both benchmarks (Jia, 2015; Odili & Mohmad Kahar, 2016).

Table 1. Benchmark characteristic and optimal solution of each dataset

Dataset	Number of cities	Optimal
att48	48	33522
st70	70	675
Eil76	76	538
pr152	152	73682
gil262	262	2378
rd400	400	15281
pr1002	1002	259045
d1291	1291	50801
fnl4461	4461	182566
brd14051	14051	469385
berlin52	52	7542
pr76	76	108159
kroa100	100	21282
eil101	101	629
ch150	150	6528
tsp225	225	3916

The performance of the proposed algorithm is evaluated based on two criteria including average distance produced by each algorithm calculated as the average Euclidean distance through all cites starting from the start city to all cites and returning to the start city; and the best solution provided by the algorithm in all number of runs to indicate which algorithm surpasses the produced minimum tour distance. For comparison, all parameter settings are fixed for all algorithms as shown in Table 2 in accordance with a previous study (Odili & Mohmad Kahar, 2016).

Table 2. Parameters of all algorithms

ABO	ACO	ABC	Hybrid BWO
Population =40	Ants =D*	Population= D*	Population=40
$\boldsymbol{m.k}$=1.0	β =0.5	ϕij= rand (−1, 1)	$N_l = 1000$
bgmax/bpmax= 0.6	ρ =0.65	ωij= rand (0, 1.5)	$pr = 0.6$
$lp1/lp2$=0.5	α =1.0	SN= NP/2	Limit max=50
$w.k$ =0.1	$Q = 200$	Limit= D^* SN	$nr = 20$
N/A	$q0 = 0.9$	Max cycle number =500	$CR = 0.44$
N/A	N/A	Colony =50	$nm = 0.4$
Total number of runs =50			

Results of the first scenario experiments are shown in Table 3 where best performances are highlighted. For each dataset, the minimum distance (best) and mean distances of the algorithms for 50 runs were recorded.

Table 3. First scenario results

Problem	Tour Length	Algorithm			
		ABO	**ACO**	**ABC**	**Hybrid BWO**
berlin52	Best	7542	7548.99	9479.11	7542
	Mean	7616	7659.31	10,390.26	**7609**
st70	Best	676	696.05	1162.12	676
	Mean	678.33	709.16	1230.49	**677.12**
eil76	Best	538	554.46	877.28	538
	Mean	563.04	561.98	931.44	**541.23**
pr76	Best	108167	115,166.66	195,198.90	108171
	Mean	**108,396**	116,321.22	205,119.61	111,126
kroa100	Best	21311	22,455.89	49,519.51	21298
	Mean	22163.8	22,880.12	53,840.03	**21832.91**
eil101	Best	640	678.04	1237.31	634
	Mean	640	693.42	1315.95	**638.73**
ch150	Best	6532	6648.51	20,908.89	6531
	Mean	**6601**	6702.87	21,617.48	6923.13
tsp225	Best	3917	4112.35	16,998.41	3920
	Mean	3982	4176.08	17,955.12	**3961.23**

Minimum tour distance has been obtained by the hybrid BWO algorithm in six datasets (75% of the datasets) followed by the ABO algorithm which managed to secure the best minimum performance in two datasets. This shows that the hybrid BWO algorithm can find better solution in the local region of the best solution by using VND. Results of the experiments in the second scenario are shown in Table 4. The performance of the proposed hybrid BWO algorithm surpasses all algorithms in eight (8) datasets while in second place is the ABO algorithm, which produces the best results in two datasets. This is again attributed to the use of VND in the BWO algorithm to solve the local optima problem

Table 4. Second scenario results

Problem	Tour Length	Algorithm				
		ABO	**PSO**	**ACO**	**HPSACO**	**Hybrid BWO**
att48	Best	33524	33734	33649	33524	33523
	Mean	33579	33982	33731	33667	**33553**
st70	Best	676	691.2	685.7	680.3	676
	Mean	**678.33**	702.6	694.7	698.6	677.12
eil76	Best	538	572.3	550.7	546.2	538
	Mean	563.04	589.1	560.4	558.1	**541.23**
pr152	Best	73730	75361	74689	74165	73722
	Mean	73990	75405	74936	74654	**73934**
gil262	Best	2378	2513	2463	2413	2378
	Mean	2386	2486	2495	2468	**2379**

rd400	Best	15301	16964	16581	16067	15298
	Mean	15304	17024	16834	16513	**15300**
pr1002	Best	259132	278923	269758	267998	259740
	Mean	**261608**	279755	271043	269789	269796
d1291	Best	50839	53912	52942	52868	50811
	Mean	50839	54104	53249	52951	**50823**
fnl4461	Best	182745	199314	192964	191352	182712
	Mean	183174	199492	194015	192585	**183123**
brd14051	Best	469835	518631	505734	498471	469745
	Mean	479085	519305	511638	503594	**478257**

The proposed hybrid BWO algorithm can find better tour distance compared to other algorithms. The algorithm iteratively looks for the local regions around the best solution found during the run. This is used as the indicator to reactivate the search process and update the algorithm with new populations. The purpose of intensifying the search process is to force the algorithm to perform deep searching. Once it converges to a local optimum, it jumps to other region in the search space, thereby increasing the probability to explore more regions of the search space. Fig. 6 displays the final comparison is performed between both the proposed hybrid BWO algorithm and VND algorithm to show the benefit of the hybridization. Results show that the proposed algorithm outperforms VND in all datasets. Note that the results of both algorithms have been normalized to produce better presentation

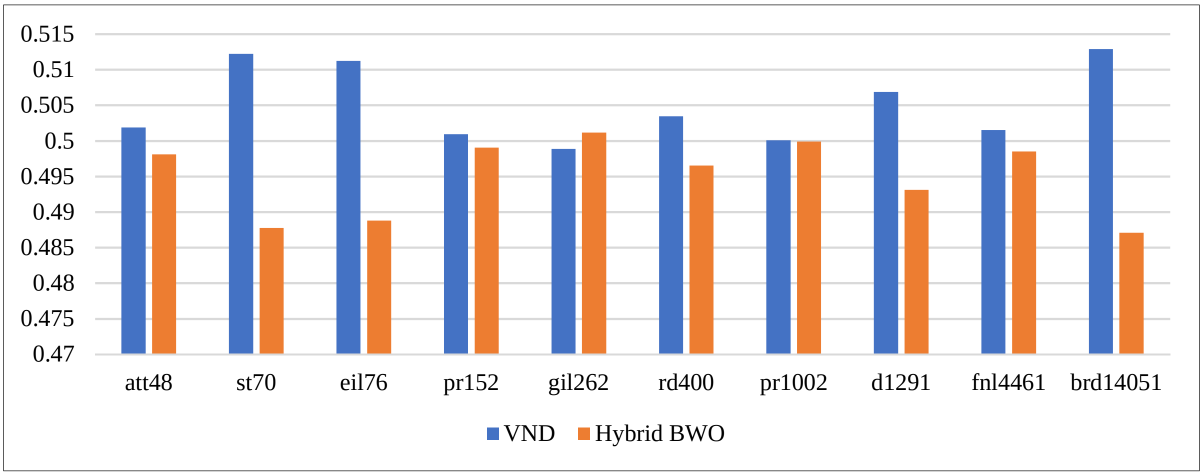

Fig. 6. VND vs. hybrid BWO algorithm

5. Conclusions

This research aims to improve the performance of the BWO algorithm in terms of finding better tour which is the minimum tour distance. The proposed hybrid BWO algorithm has solved the problem by using two modifications and this can be considered as the contribution of this study. First, the use of VND increases the probability to find more solutions with better tour distance and search deeply around the best regions that have the best solution so far. Secondly, an indicator is used to check the convergence of the results. This indicator will reactivate the search process of the algorithm by updating its memory with new populations located in different regions of the search space. The results of the experiments showed that the proposed hybrid BWO algorithm has obtained the minimum tour distances compared to ACO, ABO, ABC, PSO, and HPSACO algorithms. Both modifications have enhanced the BWO algorithm by improving the exploration and exploitation performances during the algorithm run. Despite the superior performance obtained by the BWO algorithm with the two modifications, there are two limitations that can be used as a guide for researchers who are interested in this BWO algorithm. The first one is the time-consuming process as a result of using VND, and the second is the number of parameters used in the BWO algorithm. The proposed limit parameter requires more attention in terms of optimization using adaptive and self-adaptive strategies to find the best value. Future research can also be focused on the application of the hybrid BWO algorithm in several optimization problems, such as vehicle routing, clustering, and classification with other neighborhood search algorithms.

Acknowledgements

The Ministry of Higher Education Malaysia has funded this study under the Transdisciplinary Research Grant Scheme, TRGS/1/2018/UUM/02/3/3 (S/O code 14163).

References

Abuhamdah, A. (2020). Adaptive black widow optimisation algorithm for data clustering. Journal Mathematics in Operational Research, 2002, 1–25.

Al-behadili, H. N. K., Ku-mahamud, K. R., & Sagban, R. (2019). Annealing strategy for an enhance rule pruning technique in ACO-based rule classification. Indonesian Journal of Electrical Engineering and Computer Science, 16(3), 1499–1507

Al-Behadili, H. N. K., Ku-Mahamud, K. R., & Sagban, R. (2020a). Hybrid ant colony optimization and iterated local search for rules-based classification. Journal of Theoretical and Applied Information Technology, 98(04), 657–671.

AL-Behadili, H. N. K., Ku-Mahamud, K. R., & Sagban, R. (2020b). Hybrid ant colony optimization and genetic algorithm for rule induction. Journal of Computer Science, 16(7), 1019–1028.

Al-Behadili, H. N. K., Sagban, R., & Ku-Mahamud, K. R. (2020c). Adaptive parameter control strategy for ant-miner classification algorithm. Indonesian Journal of Electrical Engineering and Informatics, 8(1), 149–162.

Dao, S. D., Abhary, K., & Marian, R. (2015). An adaptive restarting genetic algorithm for global optimization. Lecture Notes in Engineering and Computer Science, 2219, 455–459.

Das, A. K., & Pratihar, D. K. (2018). Performance improvement of a genetic algorithm using a novel restart strategy with elitism principle. International Journal of Hybrid Intelligent Systems, 15(1), 1–15.

Desale, S., Rasool, A., Andhale, S., & Rane, P. (2015). Heuristic and Meta-Heuristic Algorithms and Their Relevance to the Real World: A Survey. International Journal of Computer Engineering in Research Trends, 351(5), 2349–7084.

Duarte, A., Mladenović, N., Sánchez-Oro, J., & Todosijević, R. (2016). Variable Neighborhood Descent. Handbook of Heuristics, 1–27.

El-Ghazali Talb. (2009). METAHEURISTICS from design to implementation.

Ghany, K. K. A., AbdelAziz, A. M., Soliman, T. H. A., & Sewisy, A. A. E. M. (2020). A hybrid modified step Whale Optimization Algorithm with Tabu Search for data clustering. Journal of King Saud University - Computer and Information Sciences, 5(1), 1–8.

Goh, S. L., Kendall, G., Sabar, N. R., & Abdullah, S. (2020). An effective hybrid local search approach for the post enrolment course timetabling problem. Opsearch, 57(4), 1131–1163.

Hayyolalam, V., & Pourhaji Kazem, A. A. (2020). Black Widow Optimization Algorithm: A novel meta-heuristic approach for solving engineering optimization problems. Engineering Applications of Artificial Intelligence, 87(September 2019).

Hore, S., Chatterjee, A., & Dewanji, A. (2018). Improving variable neighborhood search to solve the traveling salesman problem. Applied Soft Computing Journal, 68, 83–91.

Hore, S., Dewanji, A., & Chatterjee, A. (2014). Design issues related to allocation of experimental units with known covariates into two treatment groups. Journal of Statistical Planning and Inference, 155, 117–126.

Hore, S., Dewanji, A., & Chatterjee, A. (2016). On Optimal Allocation of Experimental Units with Known Covariates into Multiple Treatment Groups. Calcutta Statistical Association Bulletin, 68(1–2), 69–81.

Hossain, M. Z., Akhtar, M. N., Ahmad, R. B., & Rahman, M. (2019). A dynamic K-means clustering for data mining. Indonesian Journal of Electrical Engineering and Computer Science, 13(2), 521–526.

Houssein, E. H., Helmy, B. E. din, Oliva, D., Elngar, A. A., & Shaban, H. (2020). A novel Black Widow Optimization algorithm for multilevel thresholding image segmentation. Expert Systems with Applications, (12), 114-159.

Jabbar, A. M., Ku-Mahamud, K. R., & Sagban, R. (2019a). An improved ACS algorithm for data clustering. Indonesian Journal of Electrical Engineering and Computer Science, 17(3), 1506–1515.

Jabbar, A. M., Ku-Mahamud, K. R., & Sagban, R. (2019b). Balancing exploration and exploitation in acs algorithms for data clustering. 97(16), 4320–4333.

Jabbar, A. M., Ku-Mahamud, K. R., & Sagban, R. (2019c). Modified ACS Centroid Memory for Data Clustering. Journal of Computer Science, 15(10), 1439–1449.

Jia, H. (2015). A Novel Hybrid Optimization Algorithm and its Application in Solving Complex Problem. International Journal of Hybrid Information Technology, 8(2), 1–10.

Moriguchi, K., Ueki, T., & Saito, M. (2015). An Evaluation of the Use of Simulated Annealing to Optimize Thinning Rates for Single Even-Aged Stands. International Journal of Forestry Research, 2015, 1–15.

Odili, J. B., & Mohmad Kahar, M. N. (2016). Solving the Traveling Salesman's Problem Using the African Buffalo Optimization. Computational Intelligence and Neuroscience. 3(1), 1–13.

Rasekh, A., & Osawa, N. (2020). Direct and indirect effect of cannibalism and intraguild predation in the two sibling Harmonia ladybird beetles. Ecology and Evolution, 10(12), 5899–5912.

Sathish, & Ananthapadmanabha. (2021). Improved black widow-bear smell search algorithm for optimal planning and operation of distributed generators in distribution system. Journal of Engineering, Design and Technology. 7(1), 1–21.

Sheriba, S. T., & Rajesh, D. H. (2021). Energy-efficient clustering protocol for WSN based on improved black widow optimization and fuzzy logic. Telecommunication Systems. 15(10), pages213–230.

Sicilia, J. A., Quemada, C., Royo, B., & Escuín, D. (2016). An optimization algorithm for solving the rich vehicle routing problem based on Variable Neighborhood Search and Tabu Search metaheuristics. Journal of Computational and Applied Mathematics, 291, 468–477.

Stegherr, H., Heider, M., & Hähner, J. (2020). Classifying Metaheuristics: Towards a unified multi-level classification system. 7(3), 1–17.

Stützle, T., & Hoos, H. H. (2000). MAX –MIN Ant System. Future Generation Computer Systems, 16, 889–914.

Tan, J. S., Goh, S. L., Sura, S., Kendall, G., & Sabar, N. R. (2020). Hybrid particle swarm optimization with particle elimination for the high school timetabling problem. Evolutionary Intelligence. (3). 1-16

TSPLIB. (1995). Symmetric traveling salesman problem (TSP).

Venkatesh, B., & Anuradha, J. (2019). A review of Feature Selection and its methods. Cybernetics and Information Technologies, 19(1), 3–26.

Yang, X.-S., & Yang, X.-S. (2014). Simulated Annealing. In Nature-Inspired Optimization Algorithms (pp. 67–75).

Zhou, K., Wan, W., Chen, X., Shao, Z., & Biegler, L. T. (2013). A parallel method with hybrid algorithms for mixed integer nonlinear programming. In Computer Aided Chemical Engineering, 19(1). 271-276.

AUTHOR BIOGRAPHIES

Ayad Mohammed Jabbar was born in Iraq, Basra, in 1985. He holds a Bachelor Certificate in Computer Science in 2008, and his Master degree (IT) in 2011 from Universiti Utara Malaysia. His PhD was obtained from Universiti Utara Malaysia in 2021 in Computer Science. His interests include clustering, ant colony optimization and vehicle routing problems.

Ku Ruhana Ku-Mahamud She holds a Bachelor in Mathematical Sciences and a Master's degree in Computing, both from Bradford University, United Kingdom in 1983 and 1986 respectively. Her PhD in Computer Science was obtained from Universiti Pertanian Malaysia in 1994. As an academic, her research interests include ant colony optimization, pattern classification and vehicle routing problems.