

Article

Improvement of Traveling Salesman Problem Solution Using Hybrid Algorithm Based on Best-Worst Ant System and Particle Swarm Optimization

Muhammad Salman Qamar^{1,2}, Shanshan Tu^{3,*}, Farman Ali¹, Ammar Armghan⁴, Muhammad Fahad Munir², Fayadh Alenezi⁴, Fazal Muhammad⁵, Asar Ali¹ and Norah Alnaim⁶

¹ Department of Electrical Engineering, Qurtuba University of Science and IT, Dera Ismail Khan 29050, Pakistan; mskhan131@qurtuba.edu.pk (M.S.Q.); drfarmanali.optics@qurtuba.edu.pk (F.A.); engrasar45@qurtuba.edu.pk (A.A.)

² Department of Electrical Engineering, International Islamic University, Islamabad 44000, Pakistan; fahad.munir@iiu.edu.pk

³ Engineering Research Center of Intelligent Perception and Autonomous Control, Faculty of Information Technology, Beijing University of Technology, Beijing 100124, China

⁴ Department of Electrical Engineering, College of Engineering, Jouf University, Sakaka 72388, Saudi Arabia; aarmghan@ju.edu.sa (A.A.); fshenezi@ju.edu.sa (F.A.)

⁵ Department of Electrical Engineering, University of Engineering Technology, Mardan 23200, Pakistan; fazal.muhammad@uetmardan.edu.pk

⁶ Department of Computer Science, College of Sciences and Humanities in Jubail, Imam Abdulrahman bin Faisal University, Dammam 31441, Saudi Arabia; Nmalnaim@iau.edu.sa

* Correspondence: sstu@bjut.edu.cn



Citation: Qamar, M.S.; Ali, F.; Armghan, A.; Munir, M.F.; Alenezi, F.; Muhammad, F.; Ali, A.; Alnaim, N. Improvement of Traveling Salesman Problem Solution Using Hybrid Algorithm Based on Best-Worst Ant System and Particle Swarm Optimization. *Appl. Sci.* **2021**, *11*, 4780. <https://doi.org/10.3390/app11114780>

Academic Editor: Giancarlo Mauri

Received: 25 April 2021

Accepted: 19 May 2021

Published: 23 May 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Abstract: This work presents a novel Best-Worst Ant System (BWAS) based algorithm to settle the Traveling Salesman Problem (TSP). The researchers has been involved in ordinary Ant Colony Optimization (ACO) technique for TSP due to its versatile and easily adaptable nature. However, additional potential improvement in the arrangement way decrease is yet possible in this approach. In this paper BWAS based incorporated arrangement as a high level type of ACO to upgrade the exhibition of the TSP arrangement is proposed. In addition, a novel approach, based on hybrid Particle Swarm Optimization (PSO) and ACO (BWAS) has also been introduced in this work. The presentation measurements of arrangement quality and assembly time have been utilized in this work and proposed algorithm is tried against various standard test sets to examine the upgrade in search capacity. The outcomes for TSP arrangement show that initial trail setup for the best particle can result in shortening the accumulated process of the optimization by a considerable amount. The exhibition of the mathematical test shows the viability of the proposed calculation over regular ACO and PSO-ACO based strategies.

Keywords: Particle Swarm Optimization (PSO); Best-Worst Ant System (BWAS); Ant Colony Optimization (ACO); Traveling Salesman Problem (TSP)

1. Introduction

The solution of Traveling Salesman Problem (TSP) [1] and its variants aids in finding the minimum travel distance as a cost function for a variety of computing problems. The extraordinary theoretical importance of TSP solution has a variety of practical applications, for example, design of circuit board layout [2] and transportation [3]. With the advent of intelligent transport, multi-machine industrial applications and swift delivery services, there has been an increased interest in optimizing the solution for TSP. The shortest travel distance, with the condition of traversing each node at least once, can be found by using different approaches with each having some advantage and associated weakness. Finding solution through an exact algorithm such as branch and bound [4] can take a very long run-time, particularly for a large number of nodes. Given a build-up of nodes

$y = y^1, y^2, y^3, \dots, y^n$, as the objective of finding nearest travel path solution, including as much as decreased length with one time node visit condition. Vowing to practical applications and theoretical features, it is considered NP hard problem in conjunctive optimization. The other problems associated with direct methods like branch and bound [5] is the ultra-long run-time because the size of TSP iteration increases with the characters as combination explosion, for which even dynamic programming methods can hardly provide a solution within an adequate amount of time. The other type to solve TSP are the heuristic approaches such as bat algorithm [6], fruit fly algorithm [7–9], particle swarm algorithm [10,11], cuttlefish optimization algorithm [12] and artificial bee colony algorithm [13], which are result of the continued development of artificial intelligence. These algorithms work with a goal to find a satisfactory result, regardless of the concern to find an ideal one, because of time constraint. Ant Colony Optimization (ACO) [14], which is proposed by Italian Dorigo, is a swarm intelligent optimization algorithm with wide range of application like multi-objective optimization problems [15], vehicle routing problem [16], resource constrained job scheduling [17], dynamic railway junction rescheduling [18]. The authors in [19] integrate a local search (LS) method with an existing PSO algorithm with strong global search ability, named comprehensive learning PSO (CLPSO). In [20], the authors have explained a new multiobjective programming model for the target disassembly sequencing. It proposes an improved multiobjective ant colony algorithm to derive optimal target disassembly sequences. ACO is based on the social solution path which is build by a group of ants to travel in food search and then return back with an optimized travel path [21,22]. The basic principle is the exchange of useful information about the path by the ants, and thus improving the solution quality [23]. Better paths are chosen by ants as indicated by the amount of pheromone, where then the pheromone is updated. So, at this stage through different iterations the global optimal solution is induced [24]. The improved ACO like parallel and direction guided algorithms are presented in order to resolve premature and algorithm performance of ACO impairments [25]. From the analysis of predatory behavior of birds practical swarm optimization (PSO) intelligent algorithm has been introduced, which function is to modify the location and speed of elements to gain optimal outcomes in terms of population information and experience [26,27]. Moreover, PSO can be easily implemented due to its simple structure, therefore, it is majorly applied on problems like resource allocation, wind power prediction, etc., [28–30].

The above discussed PSO and ACO algorithms present efficient performance against problems. However, the complex structure and probability of ACO has decreased its services. Therefore, low cost and simple optimization procedures are necessary to overcome on these issues. Thus, the hybrid algorithm of combined ACO and PSO is studied in this paper to solve TSP. Enormous benchmark problems are used to test the performance of ACO-PSO and performance is compared with the recently proposed variants of hybrid ACO-PSO.

Organization and Notation of Paper

The remaining part of the paper is organized as follows: PSO is elaborated in Section 3. Section 4 presents ACO. Section 2 presents the Max Min ant system. Section 5 explains the proposed modified version of the Best-Worst Ant System. Similarly, Section 6 discusses hybrid algorithm of particle swarm and ant colony. Section 7 defines results and discussion. Section 8 concludes the study.

2. Max and Min Ant System

The Max and Min Ant Colony System (MMAS), which is planned in the trial examination and use of the insect framework, makes three upgrades to the insect province framework.

1. To enhance the capacity of searching, highest value is set for every path of initial pheromone.

- In order to update pheromone, just ant with closest path is allowed in an iteration, which is measured as follow.

$$\rho_{mn}(t + 1) = \tau\rho_{mn}(t) \frac{1}{L_{best}} \tag{1}$$

where L_{best} addresses the briefest length in the current emphasis:

- To stay away from untimely assembly of the calculation, the pheromone centralization of every way (τ) is restricted to $[\tau_{min}, \tau_{max}]$ and the worth past this reach is persuasively set to τ_{min} or τ_{max} .

3. Brief Review on PSO

PSO is considered a clever algorithm which is attained from the birds life way, where the traveling particle is dependent upon the neighborhood and swarm’s global position. In the process of optimization, the number of iterations are kept fixed to ensure the convergence of the algorithm, after the selection of best number and best global value from the locals. In addition, every element is given a memory space for the best spot at any point found by utilizing the speed and position for the condition of every particle in the i th emphasis, addressed by ϑ_p and ζ_p , respectively. The values of best position for the i th particle (Pb) and best global position for the whole folk (Gb) are used to optimize the movement of the whole group. The velocity, position and learning factors of the individual particles are updated according to Equations (1)–(3), respectively.

$$\vartheta_{p+1} = \omega * \vartheta_p + c, \tag{2}$$

$$\zeta_{p+1} = \zeta_p + \vartheta_{p+1}, \tag{3}$$

$$d = d_1 * \zeta_1 * (Pb_p - \zeta_p) + d_2 * \zeta_2 * (Gb_p - \zeta_p), \tag{4}$$

where the inertial weight ω determines the total swarm population, the learning factors d_1 and d_2 determine the self-seeking ability and collective search ability of an individual particle. ζ_1 and ζ_2 can be between 0 and 1 with random probability.

4. Ant Colony Optimization

The ant colony algorithm is based on observing real abilities of ants to search for the best possible path (shortest) for food. To describe it mathematically, consider a group of x number of ants which are located at random locations across y number of cities, including primarily pheromone $\rho_{mn}(0)$ at the side of every city. The main city is added with taboo table $Z_k(0)$ of each ant, whereby, each ant finding the city to proceed the coming step using probability and can be defined as:

$$P_{mn}^k = \begin{cases} \frac{\rho_{mn}[t]^\alpha [\eta_{mn}]^\beta}{\sum_{n \in A_k} \rho_{mn}[t]^\alpha [\eta_{mn}]^\beta} & n \in A_k \\ 0 & n \notin A_k \end{cases} \tag{5}$$

where P_{mn}^k indicates the likelihood of moving from city m to city n of the k th insect and $\rho_{mn}(t)$ is for the benefit of that of the i th iteration. $\eta_{mn}(t)$ is the corresponding of the distance between city m and city n . Parameters α and β are the part of pheromones and way length in picking probabilities, separately. Urban communities outside of the taboo table comprise of the set A_t . After the taboo table of all the subterranean insect are satisfied, the all out distance every insect goes through will be determined and the most limited way will be recorded. All the while, the pheromone on each edge will be refreshed by the Equation (5) given beneath.

$$\rho_{mn}(t + 1) = \tau\rho_{mn}(t) + \Delta\rho_{mn}^k \tag{6}$$

where

$$\Delta\rho_{mn} = \sum_{k=1}^m \Delta\rho_{mn}^k \tag{7}$$

where ρ_{mn} addresses the amount of the amassed pheromones between city m and city n and k τ_{mnk} is for the benefit of the pheromone created by insect k on the edge mn . In the mean time, p is a number somewhere in the range of 0 and 1 showing the degree of pheromone dispersal. By and large, the aggregate sum of pheromone delivered in one cycle, which is addressed as Q , is restricted to speed up the intermingling speed. In the mean time, L_k is for the benefit of the all out length of the k th way. For this situation, the recipe to ascertain ρ_{mn}^k is expressed in Equation (7) as

$$\Delta\rho_{mn}^k = \frac{Q}{L_k} \tag{8}$$

5. The Proposed Best-Worst Ant System

The best worst ant system (BWAS) idea attempts to enhance the performance of ACO models applying evolutionary algorithm ideas. The suggested BWAS uses the ant system (AS) transition rule which can be stated as Equation (9):

$$P_{r,s}^k = \begin{cases} \frac{\rho_{rs}[t]^\alpha [\eta_{rs}]^\beta}{\sum_{\mu \in j_k(r)}^{\max} \rho_{r\mu}[t]^\alpha [\eta_{r\mu}]^\beta} & \text{if } s \in j_k \\ 0 & \text{if } s \notin j_k \end{cases} \tag{9}$$

ρ_{rs} being the pheromone trail of edge (r, s) , η_{rs} being the heuristic esteem, $J_k(r)$ being the set of nodes that stay to be visited by ants k , and with α and β being real value weights. In addition, the typical AS evaporation rule is utilized:

$\rho_{rs} \leftarrow (1 - \tau)\rho_{rs}, \forall r, s$, with $\tau \in [0, 1]$ belongs to the pheromone fall off parameter. Furthermore, the BWAS has the three after daemon actions that are observed in deep in [31]

Best-Worst Performance Update Rule

This performance update rule is based on the Population Based Incremental Learning (PBIL) [32] probability array update rule. The offline pheromone trail updating is given below: $\rho_{rs} \leftarrow \rho_{rs} + \Delta\rho_{rs}$, where

$$P_{r,s}^k = \begin{cases} f(C(S_{global-best})) & \text{if } (r, s) \in S_{global-best} \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

In Equation (10), the $f(C(S_{global-best}))$ represents the measure of pheromone to be placed by the global best ant, which relies upon the quality of the solution it produced, $C(S_{global-best})$. Additionally, the edges present in the worst current ant are punished: $\forall (r, s) \in S_{current-worst}$ and $(r, s) \notin S_{global-best}, \rho_{rs} \leftarrow (1 - \tau)\rho_{rs}$

6. Combined Algorithm of PSO and ACO

6.1. Strategy

Swarm intelligence based ACO-PSO are good algorithm for solving the problems optimally. On the other side TSP is also a discrete optimization problem, however, its control parameters are handled by experience technique. The essential thought of ACO-PSO is to join the upsides of the two calculations and use BWAS to advance the boundaries of ACO, which is applied to tackle TSP. With the mix of both BWAS and PSO, ACO-PSO can set boundaries inside a sensible reach, in this manner upgrading the looking through capacity and accelerating the union.

There are two instatement measures in ACO-PSO. One of the interaction is called PSO introduction. In this cycle, N particles with three boundaries α, β and τ of every individual are haphazardly created to frame a $3 \times N$ cluster. Where, $\alpha, \beta \in [0, 15]$ and τ

$\in [0, 1]$. In MMAS instatement, which is the other introduction measure, the underlying pheromone of each side $\rho_{mn}(0) = c$ what's more, pheromone variety $\Delta \rho_{mn} = 0$ are set. At that point haphazardly place N subterranean insects and add the underlying chosen city into untouchable table. The system of ACO-PSO is given in Algorithm 1.

It ought to be seen that g and G are the current cycle and the complete emphasis individually. In the interim, n represents the city number of the issue. Furthermore, to check the searching performance of the ACO-PSO, a few standard test sets are utilized to test the effectiveness of the algorithm.

Algorithm 1 Pseudocode: Hybrid ACO-PSO

PSO initialization:

For $d = 1$ to D

MMAS introduction:

While (not arrive at the most extreme cycles of MMAS)

for $i = 1$ to i

For $n = 1$ to K

Determine target city as per condition (1)

Update Taboo Table

end

end

Calculate the length of every subterranean insect way;

Find the ideal arrangement, the most exceedingly terrible arrangement and the worldwide ideal arrangement of this emphasis;

Update worldwide pheromone utilizing condition (5);

end;

Set the most brief way length as the wellness work esteem;

Update the speed and position of every particle;

end

6.2. Pheromone Trail Mutation

The pheromone trails endure mutations to announce diversity in the search, as done in PBIL with the memoristic structure. To do as such, Equation (11) shows how each line of the pheromone matrix is transformed with probability P_m as:

$$\rho' = \begin{cases} \rho_{rs} + mut(it, \rho_{threshold}), & \text{if } a = 0 \\ \rho_{rs} - mut(it, \tau_{threshold}), & \text{if } a \neq 0 \end{cases} \quad (11)$$

with a being an arbitrary value in $0, 1$, it being the present iteration, $\tau_{threshold}$ being the average of the pheromone trail in the edges forming the global best solution and with $mut(\cdot)$ being a capacity making a more grounded mutation as the iteration counter increases.

6.3. Restart of the Search Process When It Gets Stuck

The pheromone matrix is restarted by setting all its components to τ_0 when the number of edges that are distinctive between the present best and the present worst solutions is lesser than an obvious percentage. A simplified structure of a common BWAS algorithm [31] can be used to make the Algorithm 2:

Algorithm 2 Pseudocode: Best worst ant system algorithm.

Give an initial pheromone value, ρ_0 to each edge
For $n = 1$ to k do (in equal)
 Place insect n in an initial hub s and incorporate s in D_n
 While (insect n not in an objective hub) do
 Select the following hub to visit, $r \notin D_n$ by the AS change rule.
 For $D_n = 1$ to m do
 Run the nearby pursuit enhancement for the arrangement created by subterranean insect n, r_n .
 $s_{global-best} \leftarrow$ worldwide best subterranean insect visit. $s_{global-worst} \leftarrow$ current most noticeably terrible insect visit.
 Pheromone dissipation and Best-Worst pheromone refreshing.
 If (Stop Condition isn't fulfilled) go to step 2.

7. Result and Discussion

The proposed outcomes are discussed as follows: The inertia weight is kept at 0.8 and d_1 and d_2 are the learning factors which is equal to 2. Two experimental values, the optimal solution and average value of the 20 independent runs are studied in this paper to evaluate the outcomes. The two variants, PSO and ACO, are taken as comparison algorithm in order to check the performance of ACO-PSO. As is exhibited in over, the outcomes acquired by ACO-PSO is obviously superior to those of ACO and PSO among the 16 benchmark issues. To be more instinctive, we draw the way of every calculation as per the ideal arrangements utilizing the issue Pr124.

It can be demonstrated from the Tables 1 and 2 that the performance of ACO-PSO is much superior to those of simple ACO and PSO not only in the aspect of best results, however, also in worst case. Hence, it can be concluded that BWAS algorithm provides more accurate results than those of MMAS. The analysis is performed using 16 set of different functions from the test environments of IEEE Congress on Evolutionary Computation (CEC) [33]. The mean and standard analysis of Tables 1 and 2 are mentioned in Tables 3 and 4.

Figure 1a–c explain the experimental analysis for PSO (BWAS) in terms of best tour path, best cost and averaging node branching, respectively. Figure 1a. includes the outcomes of the global best tour path based on PSO (BWAS) algorithm. It is clarified that the results present satisfactory convergence, which concludes that BWAS algorithm is more feasible as compare to the current work, generating convergence with high speed. Figure 1b. explains the best possible iterative cost in terms of time. Similarly, Figure 1c. depicts the average node branching against iterative time for pr-124. Similarly, the Figure 2a–c present the performance of best tour path, best cost and averaging node branching for ACO (BWAS) based. Figure 3a–c denote the combined results of ACO-PSO (BWAS) based on best tour path, best cost and averaging node branching. These results demonstrate that the outcomes of ACO-PSO (BWAS) are better than individually ACO (BWAS) and PSO (BWAS). PSO (MMAS) algorithm results for pr-124 are mentioned in Figure 4a–c. Figure 4a. shows the better convergence based on best tour path, while Figure 4b,c represent global minimum cost and best average node branching for pr-124. Figure 5a. describes the best possible tour path for ACO (MMAS) based algorithm, the global minimum cost in terms of ACO (MMAS) algorithm is depicted in Figure 5b. Moreover, Figure 5c. consists of best average node branching for pr-124. The integrated results of ACO-PSO (MMAS) are shown in Figure 6a–c. These results present more accurate performance than single ACO (MMAS) and PSO (MMAS). Thus, It is found from the results of the BWAS proposed model are several folds efficient that other existing variants. The Table 5 summarized the performance of the proposed approach in comparison with exist literature.

Table 1. Algorithm Performance Chart.

Problem	Proposed BWAS Algo				MMAS			
	ACO		ACO-PSO		ACO		ACO-PSO	
	Best	Worst	Best	Worst	Best	Worst	Best	Worst
Att48	10,436	35,522	10,401	35,071	34,504	36,539	35,123	35,514
Berlin52	7498	8106	7441	76,091	8054	8436	7663	7750
Bier127	133,246	153,792	121,873	124,731	208,175	214,690	124,842	125,812
kroE100	24,391	26,382	21,736	23,746	36,189	37,975	23,784	24,077
Lin105	16,267	18,371	13,079	14,787	25,429	27,571	14,990	15,100
Lin318	41,998	56,728	41,268	47,583	251,618	256,438	47,585	48,205
Pr124	62,532.5	66,282	60,192.6	64,575	64,513	66,282	60,999.3	61,496
Pr107	47,236.5	49,727	45,927	48,274	68,871	76,043	46,249	46,554
Ch130	7117.1	8367	6294	6498	13,441	13,952	6473	6525
Ch150	7446.3	7728	6593	6639	16,926	17,864	6852	6886
Ei151	436.17	451	440	449	438	463	448	452
Ei176	564.96	567.1	559	560	643	691	568	570
Ei1101	627.28	682	598	647	973	1006	700	706
kroA100	23,979	17,393	20,478	21,837	34,573	39,422	22,387	23,096
kroC100	22,857	24,748	18,539	20,378	36,260	39,123	21,579	21,804
Pr144	59,644	61,837	56,939	58,237	213,303	225,108	59,443	59,727

Table 2. Algorithm Performance Chart.

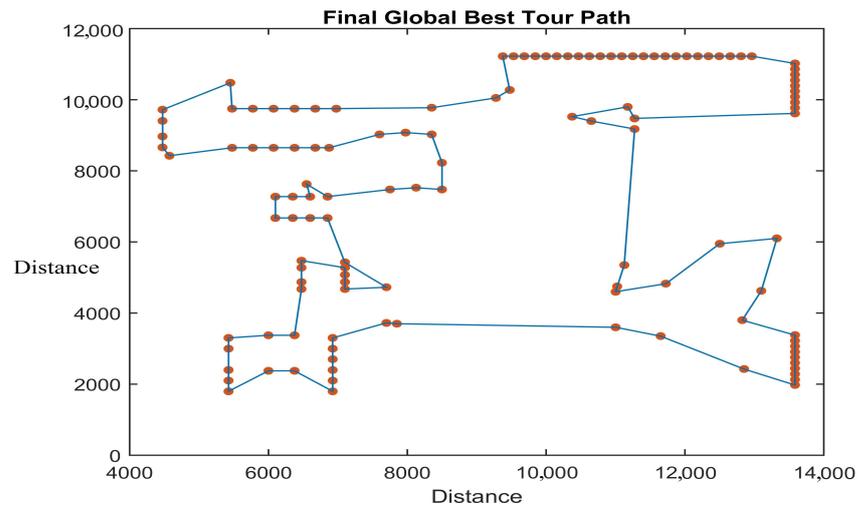
Problem	Proposed BWAS Algo				MMAS			
	PSO		ACO-PSO		PSO		ACO-PSO	
	Best	Worst	Best	Worst	Best	Worst	Best	Worst
Att48	10,398	41,135	10,368	35,071	41,307	43,477	35,123	35,514
Berlin52	7383	8906	7326	76,091	8752	9565	7663	7750
Bier127	178,926	188,165	121,873	124,731	188,651	196,774	124,842	125,812
kroE100	38,573	39,247	21,736	23,746	37,251	40,786	23,784	24,077
Lin105	26,914	27,452	13,079	14,787	28,149	28,925	14,990	15,100
Lin318	41,698	159,879	41,063	47,583	155,706	158,259	47,585	48,205
Pr124	61,319	63,726	60,192.6	64,575	61,727.2	65,436	60,999.3	61,496
Pr107	101,181	115,431	45,927	48,274	107,160	129,760	46,249	46,554
Ch130	10,054	11,038	6294	6498	11,179	12,083	6473	6525
Ch150	10,084	12,156	6593	6639	12,804	13,512	6852	6886
Ei151	410	499	443	449	511	533	448	452
Ei176	665	701	559	560	753	760	568	570
Ei1101	615	902	598	647	905	956	700	706
kroA100	35,114	38,162	20,478	21,837	37,413	39,595	22,387	23,096
kroC100	35,034	39,873	18,539	20,378	38,440	41,125	21,579	21,804
Pr144	149,376	179,837	56,939	58,237	164,662	187,458	59,443	59,727

Table 3. Mean and standard deviation of Table 1 data.

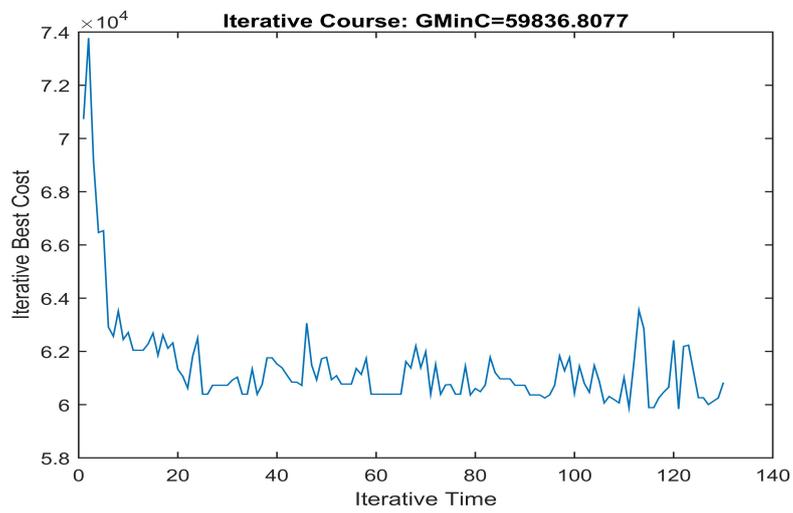
Problem	Proposed BWAS Algo				MMAS			
	ACO		ACO-PSO		ACO		ACO-PSO	
	Mean	STD	Mean	STD	Mean	STD	Mean	STD
Att48	3.9	1005.2	3.1	995	4.2	1009	3.9	999
Berlin52	16.9	502.4	15.9	410	17.5	520	16.5	499
Bier127	65.4	1115.5	64.2	995	67.2	1150	66	1005
kroE100	22.1	154	21.5	132	22.7	166	22.3	143
Lin105	31.9	2012	30.4	169	32.5	2040	31.8	201
Lin318	89.2	906.2	88.5	808	90.6	932	89.2	880
Pr124	68.5	2126	67.4	1884	69	2232	68	1934
Pr107	32.5	307	30.5	268	32.9	332	31.5	302
Ch130	76.2	209.2	74.9	200	76.8	235	75.5	210
Ch150	81	27.4	79.2	20.5	81.9	28.1	80.5	235
Ei151	81	10.2	79.5	8.4	82	11.2	81.2	99
Ei176	70.2	1.10	69.1	0.8	70.9	1.25	69.8	0.99
Ei1101	31.5	310	29.5	266	32	355	30.5	310
kroA100	22.2	309	21.5	275	22.8	350	21.9	299
kroC100	22.1	402	21.5	311	22.4	432	21.8	365
Pr144	80	1006.4	80	984	80.5	1025	80.4	1012

Table 4. Mean and standard deviation of Table 2 data.

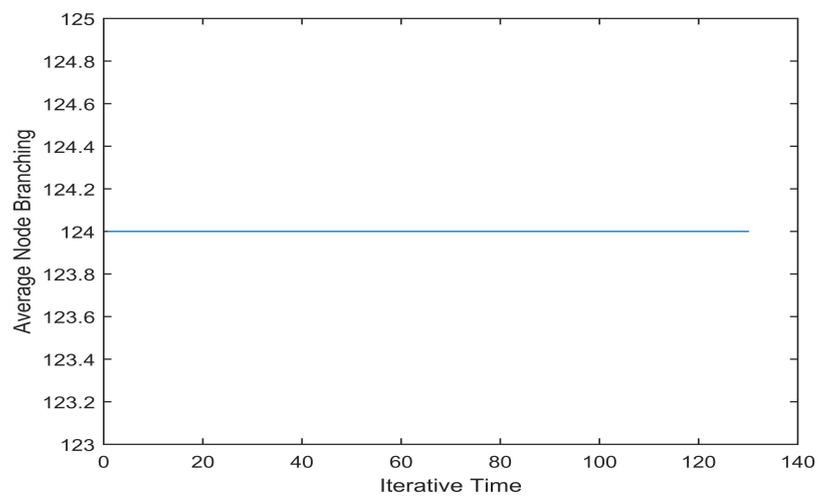
Problem	Proposed BWAS Algo				MMAS			
	PSO		ACO-PSO		PSO		ACO-PSO	
	Mean	STD	Mean	STD	Mean	STD	Mean	STD
Att48	3.2	920	2.9	890	3.9	999	3.3	992
Berlin52	15.4	410	14.9	388	16.2	455	15.5	403
Bier127	64.2	992	63.5	802	65.5	1005	64.1	920
kroE100	21.1	126	20.4	95	22.1	161	21	125
Lin105	30.5	1830	29.8	1695	31.2	2020	30.8	1725
Lin318	88.1	887	79.4	795	88.9	922	85.2	822
Pr124	66.9	1980	65.2	1910	67.8	2252	67.1	1920
Pr107	31.2	235	29.9	199	32.1	322	30.5	225
Ch130	75.1	159	73.8	102	76.2	205	74.5	155
Ch150	80.3	21.5	78.5	19.6	81.1	25.1	80.1	205
Ei151	80.5	9.4	78.4	9.1	81.4	10.4	79.9	94
Ei176	68.9	0.9	67.1	0.8	69.7	1.2	68.8	0.89
Ei1101	30.1	219	28.2	185	31.3	295	30	255
kroA100	21.3	212	20.4	175	22.1	289	21	220
kroC100	21.5	355	20.5	302	21.9	400	21	355
Pr144	78.6	958	77.9	898	79.4	992	78.4	958



(a)

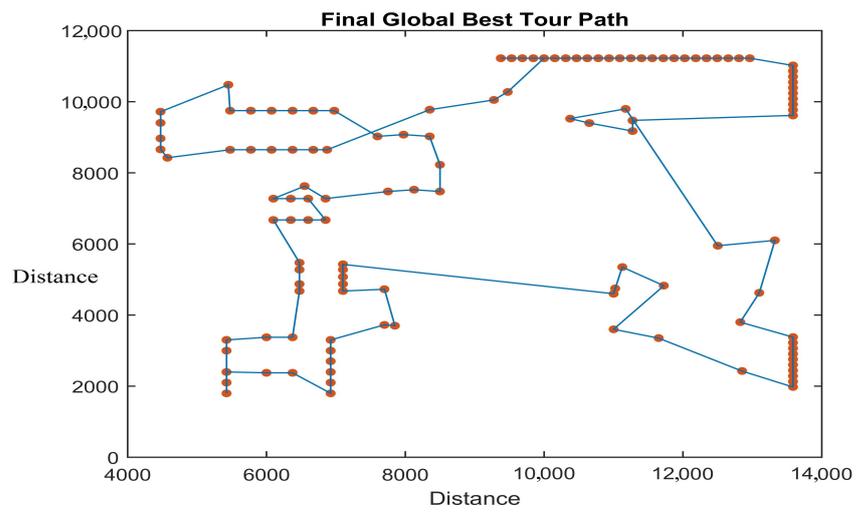


(b)

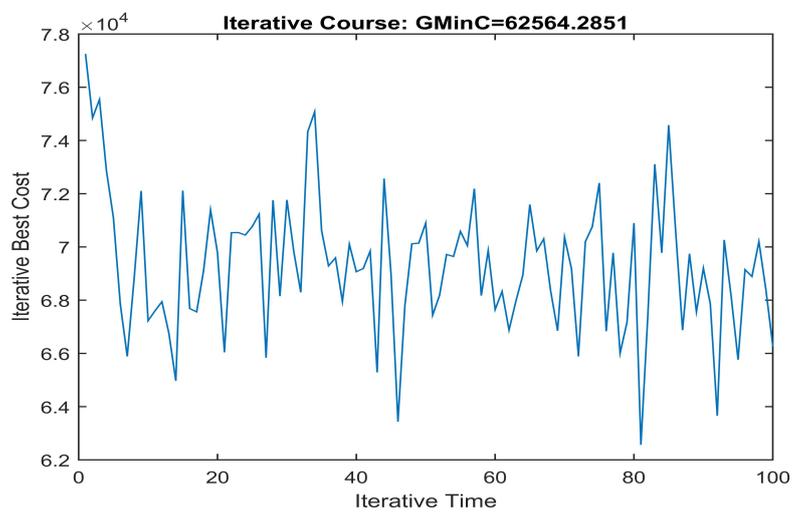


(c)

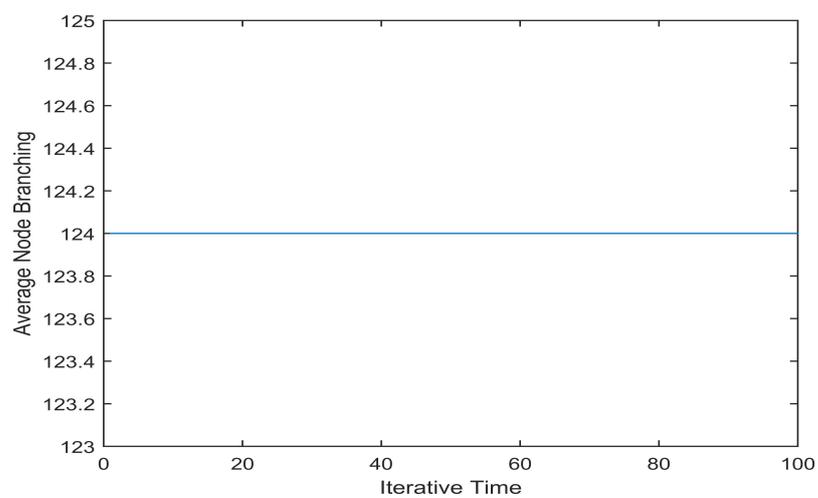
Figure 1. (a). Best tour Paths for PSO (BWAS); (b). Best cost for PSO (BWAS); (c). Average node branching for PSO (BWAS).



(a)

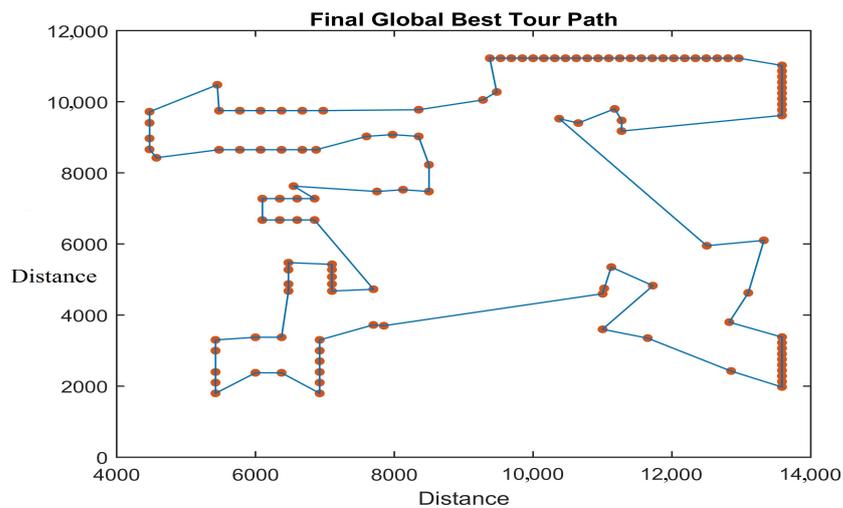


(b)

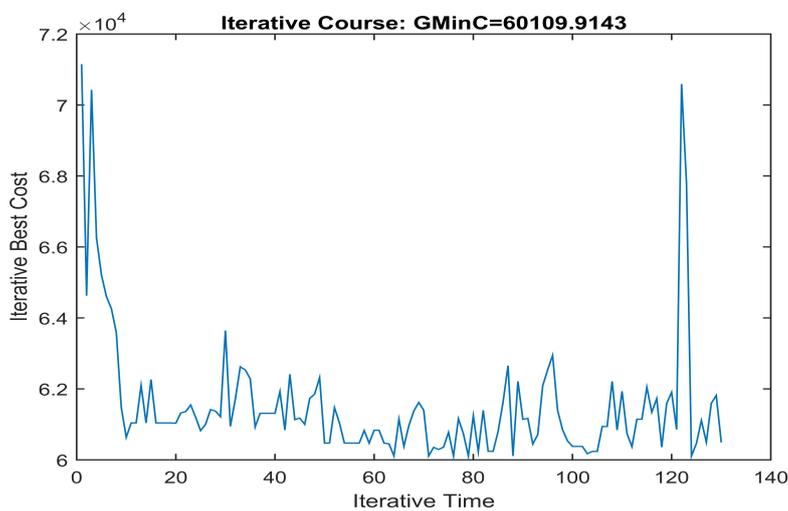


(c)

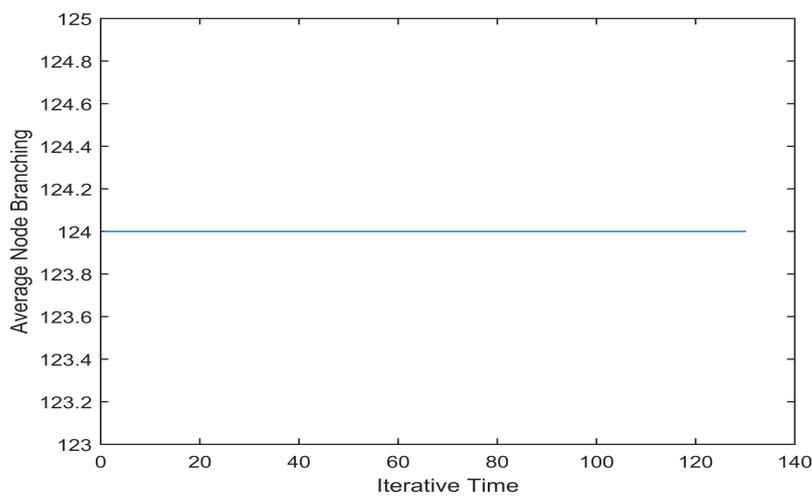
Figure 2. (a). Best tour Paths for ACO (BWAS); (b). Best cost for ACO (BWAS); (c). Average node branching for ACO (BWAS).



(a)

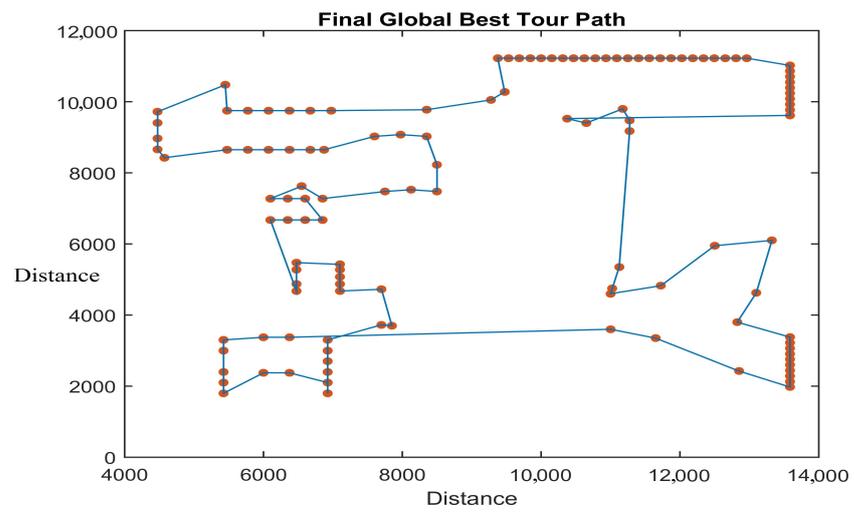


(b)

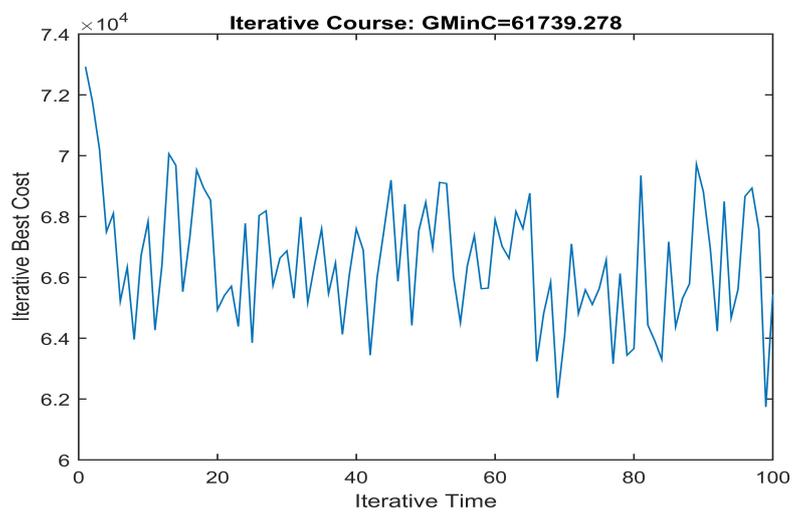


(c)

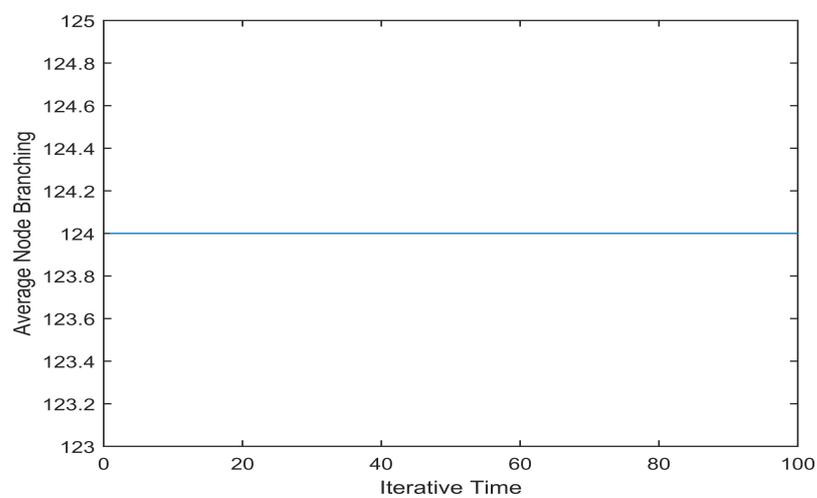
Figure 3. (a). Best tour Paths for ACO-PSO (BWAS); (b). Best cost for ACO-PSO (BWAS); (c). Average node branching for ACO-PSO (BWAS)



(a)

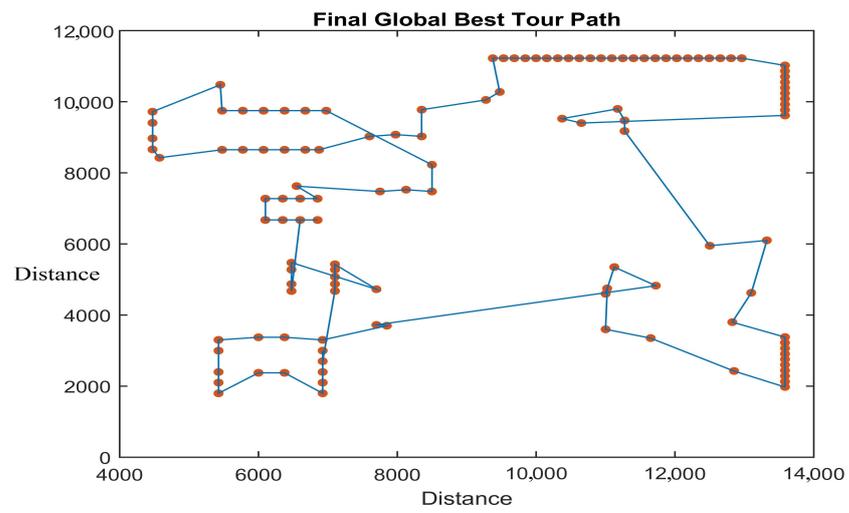


(b)

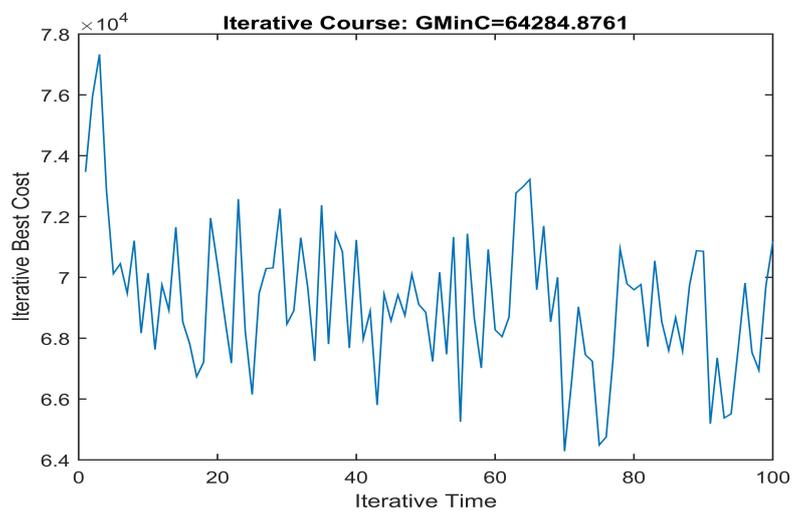


(c)

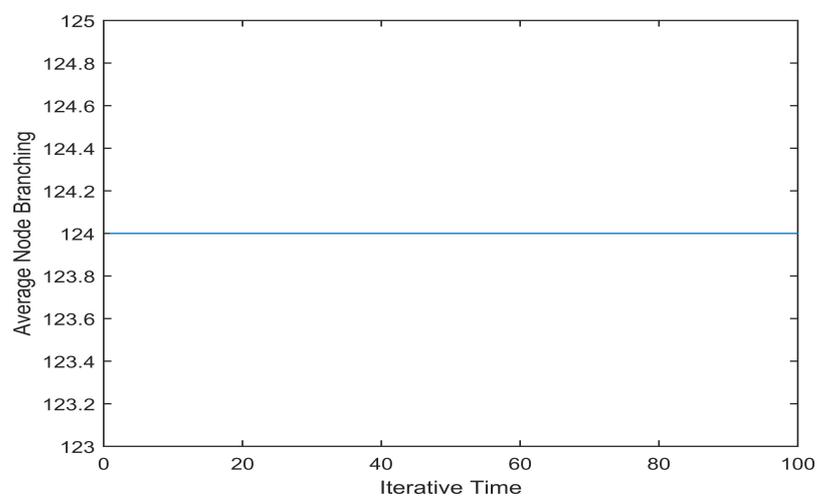
Figure 4. (a). Best tour Paths for PSO (MMAS); (b). Best cost for PSO (MMAS); (c). Average node branching for PSO (MMAS).



(a)

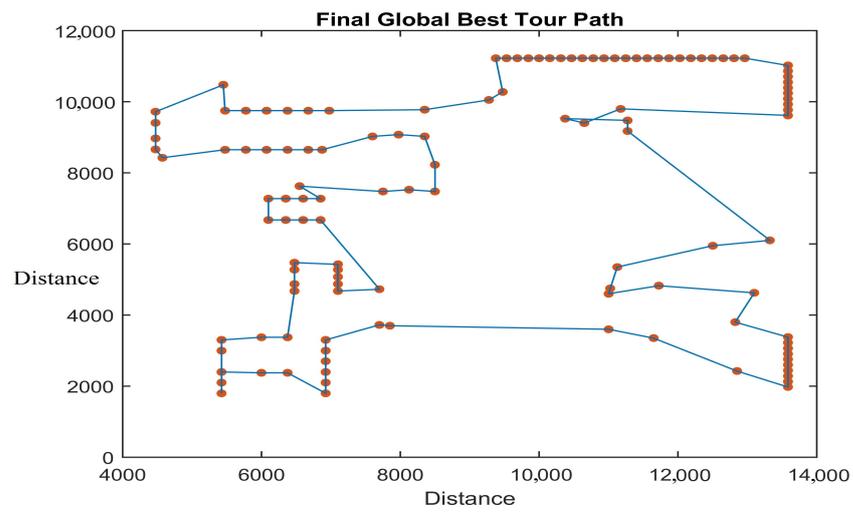


(b)

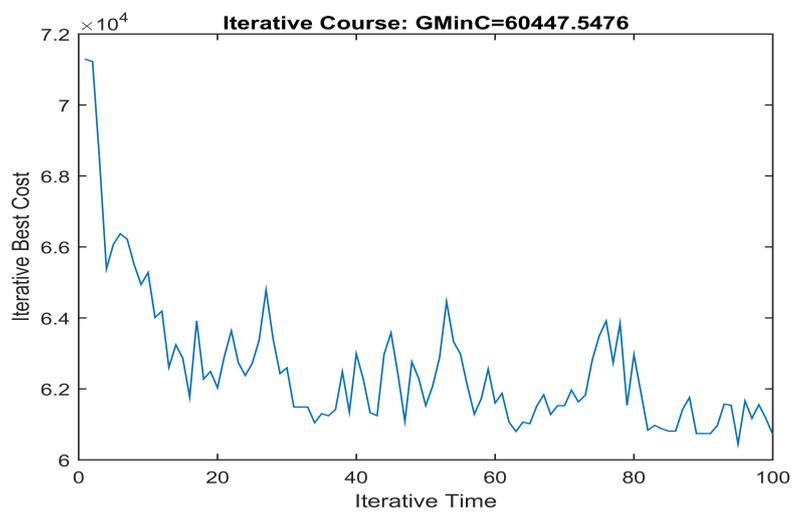


(c)

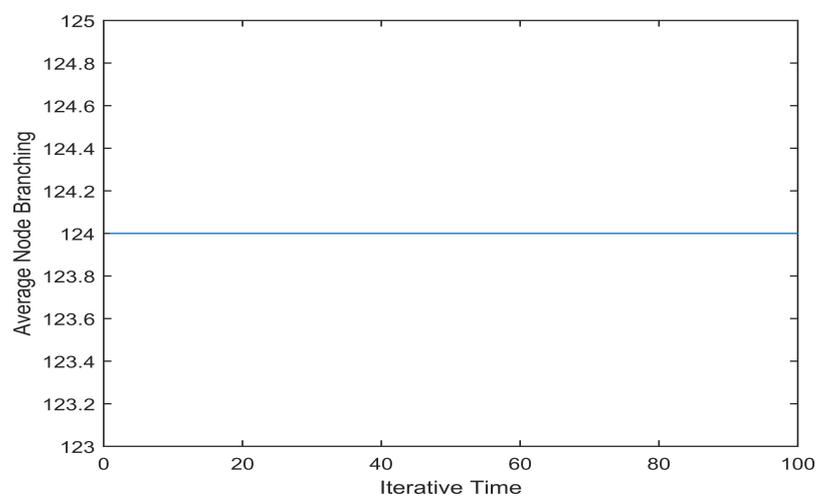
Figure 5. (a). Best tour Paths for ACO (MMAS); (b). Best cost for ACO (MMAS); (c). Average node branching for ACO (MMAS).



(a)



(b)



(c)

Figure 6. (a). Best tour Paths for ACO-PSO (MMAS); (b). Best cost for ACO-PSO (MMAS); (c). Average node branching for ACO-PSO (MMAS).

Table 5. Comparison of the proposed framework with literatures.

Instance	[26]	[27]	Proposed Model
Att48	-	10,628	10,436
Brlin52	7542	-	7498
Lin318	42,020	-	41,998
Eil101	629	629	627

8. Conclusions

This paper concludes the results of simple ACO and PSO for BWAS and MMAS algorithms. The outcomes based on simple ACO and PSO show that approaches are limited to solve TSP issues. Thus, a hybrid algorithm based on integration of BWAS based ACO and PSO is proposed, to improve the solution for TSP. The optimization of BWAS parameters is performed in concurrence with output from PSO, and results show a reasonable improvement in shortening the optimized solution path. Hence, in order to increase the searching ability and improving the overall results, multiple standard sets are used as a baseline to compare the performance of the proposed algorithm with conventional PSO and ACO methods. The results show the superiority of the proposed technique in solving TSP.

Author Contributions: Conceptualization, M.S.Q., F.A. (Farman Ali), A.A. (Ammar Armghan) and F.M.; Methodology, M.S.Q., A.A. (Ammar Armghan), F.A. (Fayadh Alemezi), M.F.M.; Validation, A.A. (Asar Ali), N.A.; Writing—Original Draft Preparation, M.S.Q., F.A. (Farman Ali); Writing—Review & Editing, S.T.; Supervision, F.A. (Farman Ali); Funding Acquisition, S.T. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported in part by the Beijing Natural Science Foundation (No. 4212015), Natural Science Foundation of China (No. 61801008), China Ministry of Education-China Mobile Scientific Research Foundation (No. MCM20200102), China Postdoctoral Science Foundation (No. 2020M670074), Beijing Municipal Commission of Education Foundation (No. KM201910005025), China National Key Research and Development Program (No. 2018YFB0803600).

Conflicts of Interest: The authors declare no conflict of interest.

References

- Osaba, E.; Del Ser, J.; Sadollah, A.; Bilbao, M.N.; Camacho, D. A discrete water cycle algorithm for solving the symmetric and asymmetric traveling salesman problem. *Appl. Soft Comput.* **2018**, *71*, 277–290. [[CrossRef](#)]
- Ghiani, G.; Manni, E.; Thomas, B.W. A comparison of anticipatory algorithms for the dynamic and stochastic traveling salesman Problem. *Transp. Sci.* **2016**, *46*, 374–387. [[CrossRef](#)]
- Hore, S.; Chatterjee, A.; Dewanji, A. Improving variable neighborhood search to solve the traveling salesman problem. *Appl. Soft Comput.* **2018**, *68*, 83–91. [[CrossRef](#)]
- Ezugwu, A.E.S.; Adewumi, A.O. Discrete symbiotic organisms search algorithm for travelling salesman problem. *Expert Syst. Appl.* **2017**, *87*, 70–78. [[CrossRef](#)]
- Rahman, M.A.; Ma, J. *Probe Machine Based Consecutive Route Filtering Approach to Symmetric Travelling Salesman Problem*; Shi, Z., Pennartz, C., Huang, T., Eds.; Springer: Cham, Switzerland, 2018; Volume 539, pp. 378–387.
- Yin, L.; Li, X.; Gao, L. A new improved fruit fly optimization algorithm for traveling salesman problem. In Proceedings of the 2016 Eighth International Conference on Advanced Computational Intelligence (ICACI), Chiang Mai, Thailand, 14–16 February 2016; pp. 21–28.
- Li, H.; Chen, J.; Huang, Q. An improvement of fruit fly optimization algorithm for solving traveling salesman problems. In Proceedings of the IEEE International Conference on Information and Automation, Hailar, China, 28–30 July 2014; pp. 620–623.
- Iscan, H.; Gunduz, M. An application of fruit fly optimization algorithm for traveling salesman problem. *J. Procedia Comput. Sci.* **2017**, *111*, 58–63. [[CrossRef](#)]
- Marinakos, Y.; Marinaki, M. A hybrid multi-swarm particle swarm optimization algorithm for the probabilistic traveling salesman problem. *Jouenal Comput. Oper. Res.* **2010**, *37*, 432–442. [[CrossRef](#)]
- Gao, S.; Han, B.; Wu, X. Solving traveling salesman problem by hybrid particle swarm optimization algorithm. *J. Control. Decis.* **2015**, *19*, 1486–1489.

11. Riffi, M.E.; Bouzidi, M. Discrete cuttlefish optimization algorithm to solve the travelling salesman problem. In Proceedings of the 2015 Third World Conference on Complex Systems (WCCS), Marrakech, Morocco, 23–25 November 2015; pp. 1–6.
12. Ismkhan, H. Effective heuristics for ant colony optimization to handle large-scale problems. *J. Swarm Evol. Comput.* **2017**, *32*, 140–149. [[CrossRef](#)]
13. Zhong, Y.; Lin, J.; Wang, L.; Zhang, H. Hybrid discrete artificial bee colony algorithm with threshold acceptance criterion for traveling salesman problem. *Inf. Sci.* **2017**, *421*, 70–84. [[CrossRef](#)]
14. Osaba, E.; Yang, X.S.; Diaz, F.; Lopez-Garcia, P.; Carballedo, R. An improved discrete bat algorithm for symmetric and asymmetric Traveling Salesman Problems. *Eng. Appl. Artif. Intell.* **2016**, *48*, 59–71. [[CrossRef](#)]
15. Saji, Y.; Riffi, M.E. A novel discrete bat algorithm for solving the travelling salesman problem. *Neural Comput. Appl.* **2016**, *27*, 1853–1866. [[CrossRef](#)]
16. Kóczy, L.T.; Földesi, P.; Tüű-Szabó, B. Enhanced discrete bacterial memetic evolutionary algorithm—An efficacious metaheuristic for the traveling salesman optimization. *Inf. Sci.* **2018**, *460*, 389–400. [[CrossRef](#)]
17. Ouenniche, J.; Ramaswamy, K.; Gendreau, M. A dual lpcal researhework for combinatorial optimization problems with TSP application. *J. Oper. Res. Soc.* **2017**, *68*, 1377–1398. [[CrossRef](#)]
18. Yin, D.; Du, S.; Wang, S. A direction-guided ant colony optimization method for extraction of urban road information from very-high-resolution images. *J. IEEE J. Sel. Top. Appl. Earth Obs. Remote. Sens.* **2016**, *8*, 4785–4794. [[CrossRef](#)]
19. Cao, Y.; Zhang, H.; Li, W.; Zhou, M.; Zhang, Y.; Chaovaitwongse, W.A. Comprehensive Learning Particle Swarm Optimization Algorithm With Local Search for Multimodal Functions. *IEEE Trans. Evol. Comput.* **2019**, *23*, 718–731. [[CrossRef](#)]
20. Feng, Y.; Zhou, M.; Tian, G.; Li, Z.; Zhang, Z.; Zhang, Q.; Tan, J. Target Disassembly Sequencing and Scheme Evaluation for CNC Machine Tools Using Improved Multiobjective Ant Colony Algorithm and Fuzzy Integral. *IEEE Trans. Syst. Man Cybern. Syst.* **2019**, *49*, 2438–2451. [[CrossRef](#)]
21. Xu, Z.; Wang, Y.; Li, S.; Liu, Y.; Todo, Y.; Gao, S. Immune algorithm combined with estimation of distribution for traveling salesman problem. *IEEJ Trans. Electr. Electron. Eng.* **2016**, *11*, S142–S154. [[CrossRef](#)]
22. Zhong, Y.; Lin, J.; Wang, L.; Zhang, H. Discrete comprehensive learning particle swarm optimization algorithm with Metropolis acceptance criterion for traveling salesman problem. *Swarm Evol. Comput.* **2018**, *42*, 77–88. [[CrossRef](#)]
23. Wang, X.; Deng, Y.; Duan, H. Edge-based target detection for unmanned aerial vehicles using competitive Bird Swarm Algorithm. *J. Aerosp. Sci. Technol.* **2018**, *78*, 708–720. [[CrossRef](#)]
24. Kim, J.J.; Lee, J.J. Trajectory Optimization with Particle Swarm Optimization for Manipulator Motion Planning. *J. IEEE Trans. Ind. Informatics* **2017**, *11*, 620–631. [[CrossRef](#)]
25. Lu, N.; Liu, Y. Application of support vector machine model in wind power prediction based on particle swarm optimization. *J. Discret. Contin. Dyn. Syst.* **2017**, *8*, 1267–1276. [[CrossRef](#)]
26. Hammed, K.; Ghauri, S.A.; Qamar, M.S. Biological inspired stochastic optimization technique (PSO) for DOA and amplitude estimation of antenna arrays signal processing in RADAR communication system. *J. Sens.* **2016**, *2016*, 9871826. [[CrossRef](#)]
27. Yang, Y.; Liu, S.; Luo, L. Hybrid Lion Swarm Optimization Algorithm for Solving Traveling Salesman Problem. *J. Phys.* **2020**, *1550*, 032027. [[CrossRef](#)]
28. Cheng, C.Y.; Chen, Y.Y.; Chen, T.L. Using a hybrid approach based on the particle swarm optimization and ant colony optimization to solve a joint order batching and picker routing problem. *J. Int. J. Prod. Econ.* **2018**, *170*, 805–814. [[CrossRef](#)]
29. Jian, C.; Li, M.; Kuang, X. Edge cloud computing service composition based on modified bird swarm optimization in the internet of things. *Clust. Comput.* **2018**, *12*, 1–9. [[CrossRef](#)]
30. Xu, C.; Yang, R. Parameter estimation for chaotic systems using improved bird swarm algorithm. *Mod. Phys. Lett. B Condens. Matter Phys. Stat. Phys. Appl. Phys.* **2017**, *31*, 15. [[CrossRef](#)]
31. Qian, H.; Su, T. Hybrid algorithm based on max and min ant system and particle swarm optimization for solving TSP problem. In Proceedings of the 2018 33rd Youth Academic Annual Conference of Chinese Association of Automation (YAC), Nanjing, China, 18–20 May 2018; pp. 683–687.
32. Zhang, L.; Bao, Q.; Fan, W.; Cui, K.; Xu, H.; Du, Y. An improved particle filter based on bird swarm algorithm. In Proceedings of the 2017 10th International Symposium on Computational Intelligence and Design (ISCID), Hangzhou, China, 9–10 December 2017; pp. 198–203.
33. Maućecj, M.S.; Brest, J. A review of the recent use of Differential Evolution for Large-Scale Global Optimization: An analysis of selected algorithms on the CEC 2013 LSGO benchmark suite. *J. Swarm Evol. Comput.* **2019**, *50*, 100428. [[CrossRef](#)]