



You have downloaded a document from
RE-BUŚ
repository of the University of Silesia in Katowice

Title: The Pheromone-Based Harmony Search Algorithm for the Asymmetric Traveling Salesman Problem

Author: Krzysztof Szwarc, Urszula Boryczka

Citation style: Szwarc Krzysztof, Boryczka Urszula. (2020). The Pheromone-Based Harmony Search Algorithm for the Asymmetric Traveling Salesman Problem. „Applied Sciences (Basel)” (2020, iss. 18, art. no. 6422), DOI: 10.3390/app10186422



Uznanie autorstwa - Licencja ta pozwala na kopiowanie, zmienianie, rozprowadzanie, przedstawianie i wykonywanie utworu jedynie pod warunkiem oznaczenia autorstwa.



UNIwersYTET ŚLĄSKI
W KATOWICACH



Biblioteka
Uniwersytetu Śląskiego



Ministerstwo Nauki
i Szkolnictwa Wyższego

Article

The Pheromone-Based Harmony Search Algorithm for the Asymmetric Traveling Salesman Problem

Krzysztof Szwarc *  and Urszula Boryczka 

Institute of Computer Science, University of Silesia in Katowice, ul. Będzińska 39, 41-200 Sosnowiec, Poland; urszula.boryczka@us.edu.pl

* Correspondence: krzysztof.szwarc@us.edu.pl

Received: 15 August 2020; Accepted: 9 September 2020; Published: 15 September 2020



Abstract: This paper presents a modification of the Harmony Search algorithm (HS) adjusted to an effective solving of instances of the Asymmetric Traveling Salesman Problem. The improvement of the technique spans the application of a pheromone, which, by serving the role of long-term memory, enables the improvement of the quality of determined results, especially for tasks characterized by a significant number of vertices. The publication includes the results of tests that suggest the achievement of effectiveness improvement through the modification of the HS and recommendations concerning the proper configuration of the algorithm.

Keywords: harmony search; pheromone-based harmony search; asymmetric traveling salesman problem; metaheuristic

1. Introduction

Due to the considerable complexity of many problems occurring in business practice, the use of classical deterministic algorithms that enable determination of exact solutions becomes impossible in an acceptable time. For this reason, approximate methods are increasingly used, among which one can distinguish metaheuristics. These algorithms have found application in the optimization of logistic processes, e.g., Nondominated Sorting Genetic Algorithm II and Multiobjective Particle Swarm Optimization were used for sustainable ship routing and scheduling with draft restrictions [1], Non-Dominated Sorting Genetic Algorithm II was applied for the design of the Shuttle-Based Storage and Retrieval System [2] and Particle Swarm Optimization-Composite Particle was used for sustainable integrated dynamic ship routing and scheduling optimization [3], as well as sustainable maritime inventory routing problem with time window constraints [4].

The Harmony Search method (HS) is an interesting metaheuristic, which despite causing a great many controversies (it is falsely [5] accused of a lack of innovativeness and similarity to the Evolution Strategies (ES) [6]), found application in the process of solving many utilitarian problems (it was applied, among others, in the process of flood control management [7], in the fourth-party logistics routing problems [8] and in the optimization of allocation of containers around a harbor [9]). The character of the technique, implying its preferential usage to the continuous space, imposes the application of sophisticated approaches aimed at adjusting the algorithm to combinatorial optimization problems, at the same time influencing the appearance of difficulties connected with an effective designing of a method of the HS adjusted to solving a significant number of practical problems.

One of the most tested concepts of combinatorial optimization is the Traveling Salesman Problem (TSP). It is characterized by unquestionable utilitarian meaning—its asymmetric version (ATSP) models the linear infrastructure located in urbanized areas, thus becoming groundwork for the planning of routes of vehicles collecting municipal waste [10] and waste electrical and electronic equipment [11]. The use of ATSP goes beyond the area of reverse logistics (that can be defined as ‘the

process of planning, implementing, and controlling the efficient, effective inbound flow, inspection and disposition of returned products and related information for the purpose of recovering value' [12]) and can be used to solve various problems occurring in distribution logistics, e.g., the distribution process of perishable products [13], post delivery service, fast food delivery or school bus operation [14].

Utilitarian importance of ATSP and HS features influenced the onset of research on the adjustment of the HS to the ATSP. Paper [15] presents suggestions of modification of the HS, owing to which the method enables defining relatively good solutions for the issue in question. In the quoted publication, however, a noticeable decline of effectiveness of the technique for instances of the problem described by the means of more than 71 vertices was emphasized.

In the process of solving the TSP substantial effectiveness is ascribed to techniques that base on analogy to the process of searching for food by ants, using an information exchange pheromone [16,17]. Due to the lack of long-term memory in the suggested approach to designing the HS, the key element of Ant Systems (AS) was adjusted to the needs of the tested algorithm. The need to use long-term memory can be argued with the results presented by [18], according to which the elimination of the memory reset mechanism made it possible to improve the results for larger ATSP instances (worsening them at the same time for smaller ones, pointing to the need to support HS memory). The aim of this article is to suggest a long-term memory mechanism that would support the HS techniques suggested in [15] and to recommend the values of parameters steering the designed modification. As a result, we want to propose a novelty combination of AS and a special HS variant, which has been adapted to solving ATSP instances. It is supposed to eliminate the imperfections of the analyzed HS.

The article comprises the following sections: the first one constitutes an introduction to the mentioned issues, the second one touches upon the classical HS, the third one describes the ATSP, the fourth one touches upon the issue of the HS hybridization, the fifth one includes the characteristic of the Pheromone-based Harmony Search (PBHS), the sixth one concentrates on the research methodology, the seventh offers a closer look at the results of tests and finally the eighth concentrates on the conclusions and recommending further work.

2. A Classical Harmony Search Algorithm

According to [19] we can assume that the aesthetic quality of a musical instrument is determined by its pitch (that in Musical Instrument Digital Interface is often represented as a numerical scale), timbre and amplitude. Timbre is determined by the harmonic content that depends on the pitch or frequency range of the particular instrument. When we adjust the pitch, we are trying to change the frequency.

Zong Woo Geem in [20] suggested a controversial metaheuristic that was based on the similarity of the process of jazz improvisation to looking for global optimum through the means of algorithmic methods. The technique assumes the existence of a structure *HM* (referred to as the Harmony Memory), which stores *HMS* harmonies that consist of a set number of pitches (representing the values of decision variables of a given result). The particular *HM* elements are interpreted as a complete solution of the problem, the value of the objective function of which is defined on the basis of its component parts.

The initial content of the Harmony Memory is generated randomly, after which it is subjected to sorting on the basis of appropriate objective function values (in such a way as to make it possible that the result in the first position can be characterized by the best result). The realization of the described steps brings about the launch of an iterative creation of subsequent solutions.

The procedure of creating a new solution applies the knowledge stored in *HM* and is based on an analogy to the process of harmony improvisation in music. The construction of such a solution consists in an iterative selection of the next pitch, according to two parameters—*HMCR* (Harmony Memory Consideration Rate) and *PAR* (Pitch Adjustment Rate). On the basis of probability *HMCR* the pitch *i* is selected, using the values located in position *i* in harmonies that belong to *HM* (otherwise random generation of an acceptable value takes place). Creating a solution based on a component *HM* a modification of the pitch with the set probability *PAR* may occur (the change of value takes place on

the basis of parameter bw , the value of which depends on the representation of the problem). To sum up the random selection of the pitch value is done with probability $1 - HMCR$, using an unmodified value from HM may occur with probability $HMCR \cdot PAR$, whereas modified value from HM is chosen with probability $HMCR \cdot (1 - PAR)$.

After generating the subsequent solution, comparison of its objective function value with the appropriate parameter describing the component located in the last position in HM is held. In the situation of determining a better result, it replaces the worst result stored in the Harmony Memory and another sorting of the HM elements is conducted.

The procedure of generating a new solution is conducted by IT iterations, after which the return of the best result occurs (located in the first position in the Harmony Memory). Pseudocode of the HS was presented in Algorithm 1.

Algorithm 1 The Harmony Search pseudocode (based on [15]).

```

1: iterations = 0
2: for i = 0; i < HMS; i ++ do
3:   HM[i]=stochastically generate feasible solution
4: end for
5: Sort HM
6: while iterations < IT do
7:   for i = 0; i < n; i ++ do
8:     Choose random r ∈ (0,1)
9:     if r < HMCR then
10:      H[i]=choose randomly available pitch on position i in HM
11:      Choose random k ∈ (0,1)
12:      if k < PAR then
13:        α = bw · random ∈ (-1,1)
14:        H[i]=H[i] + α
15:      end if
16:     else
17:      H[i]=choose randomly available pitch
18:     end if
19:   end for
20:   if f(H) is better than f(HM[HMS - 1]) then
21:     HM[HMS - 1] = H
22:     Sort HM
23:   end if
24:   iterations ++
25: end while
26: return HM[0]

```

The existence of a visible similarity between the HS and the ES caused a great deal of controversy (e.g., [6]), however in the reference article by [5] a crucial difference in the means of functioning of exploration and exploitation, present in the quoted techniques, is emphasized. The HS has three operators, the occurrence frequency of which is controlled by the values of particular parameters describing the algorithm, whereas the strategy $(\mu + 1)$ bases its functioning only upon two obligatory operations - crossover and mutation.

3. Formulating Asymmetric Traveling Salesman Problem

Based on article [21] the following definition of the TSP was adapted: for a directed graph $G = (V, Ar)$, with arc weights c_{ij} ($i, j \in \{1, 2, \dots, n\}$), a route is being searched (directed cycle including all n cities) characterized by a minimum length. The asymmetric version assumes the possibility of occurrence of inequality $c_{ij} \neq c_{ji}$.

The decision variable x_{ij} , represented by the presence of an edge between the vertex i and j in the selected solution, adapts the following values:

$$x_{ij} = \begin{cases} 1 & \text{if edge } (i, j) \text{ is part of the route,} \\ 0 & \text{otherwise.} \end{cases} \quad (1)$$

The objective function, which assumes the minimization of the values of edges included in the solution, was formulated as:

$$\sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij} \rightarrow \min. \quad (2)$$

Limiting conditions—ensuring that the salesman visits every city on his way exactly one time—were presented in the following way:

$$\sum_{i=1}^n x_{ij} = 1, \quad j = 1, \dots, n, \quad \sum_{j=1}^n x_{ij} = 1, \quad i = 1, \dots, n. \quad (3)$$

In order to avoid the occurrence of solutions representing separate cycles, instead of a connected one, it is crucial to introduce additional limitations called Miller, Tucker and Zemlin (MTZ):

$$1 \leq u_i \leq n - 1, \quad u_i - u_j + (n - 1)x_{ij} \leq n - 2, \quad i, j = 2, \dots, n. \quad (4)$$

MTZ assumes the use of u_i variables to determine the order in which each vertex i is visited by salesman during the trip. According to it, the numbers from 1 to n are being assigned to the vertices in such a way that the numbering corresponds to the order of the vertices in the route.

4. Harmony Search Hybridization

The adaptation of a given metaheuristic approach to solve a given optimization problem is a kind of a metaphor, which brings about a suitable reaction of the system to the conditions of the changing environment, which, in turn, is the surrounding of the analyzed problem. In order to prepare a suitable ability of fast reaction in approaches solving difficult optimization problems, a great many scientists aim at creating new algorithms, which cope with the difficulties of limitations, or the multitude of the analyzed dependencies between data. All the factors contribute to a snowballing increase in the number of various metaheuristic approaches. However, it is not always that new solutions skillfully cope with finding the best solution. Usually, there is a tendency to create a general approach that would correspond with a broad range of applications. However, knowing the *No Free Lunch* theorem (according to it, there is no universal, best optimization algorithm for all tasks, the theorem is described in the paper [22]), we should firmly state that it is impossible. As a result, approaches spanning several metaheuristics into one, referred to as hybridization, are used. Such a connection can have a homogeneous or non-homogeneous character and as a result, it can be realized in a sequential or parallel way. Most frequently, however, a hybrid of optimization approaches is created as a consequence of a lack of some mechanisms, essential to solve a given problem and simultaneously absent in the homogeneous approach.

Looking at hybridization in the HS one might be sensitive to the following issues: lack of a visible mechanism of machine learning (or lack of experience) during solving a given optimization problem, lack of a clear criterion of accepting worse solutions, or finally lack of official local search/greedy algorithm, which would enable fast convergence of the hybrid algorithm. Owing to that, in this part of our work we will concentrate on the possibility of connecting the HS with other techniques copying nature. The following will be briefly discussed: Ant Colony Optimization (ACO), Neural Networks (NN), Particle Swarm Optimization (PSO), Genetic Algorithm (GA) and others.

One of the approaches to the HS and the ACO is the solution of the problem connected with an optimal installation location of structural dampers [23]. Two ideas stemming from the ACO were added to the HS: pheromone trail and some value of the heuristic function. The information was automatically ascribed to every component of the solution. Owing to such an approach to the process of obtaining a better solution over a short period of time was accelerated.

A hybrid connection of the HS with the NN took place in the problem of selecting a proper proportion while producing industrial concrete mix [24]. The optimization of the suggested mix is obtained owing to the refining of this mix through steering the proportions using Neural Networks.

The connection of the HS with an immune system can be analyzed on the basis of the problem of scheduling the work of nurses, which belongs to the NP-complete class. Owing to such an approach a proper balance between local and global search was enabled and simultaneously premature convergence was prevented.

In paper [25] a new approach was described - a connection of the HS with the Cuckoo Search (CS) in the 0-1 knapsack problem. Similarly, as in the previously analyzed article, the Global-best HS (GHS) was responsible for exploration and the CS for the exploitation of the solution space. The hybrid CSGHS was definitely better than the binary version of the CS.

In the Optimal Power Flow Problem, the BCGAs-HSA methodology was suggested in [26], which connected the binary coding of the GA with the HS algorithm. In both cases for the GA and the HS, the hybrid yielded better results in every instance of the analyzed problem.

The most complicated hybrid was applied to the truss structures problem, where the HS was connected with the PSO and the ACO [27]. In this approach, the PSO with the passive congregation was applied. The PSOPC was used for global search and the ACO was applied to updating the position of particles searching the solution space. The Fly-back mechanism was also used here. The above hybrid was compared with various variants of the PSO.

It can be noticed that more and more often [28] the HS is treated, with some modifications taken from the Swarm Intelligence, as a technique of global search, hence there are numerous modifications connected with the local search. The missing learning mechanism is located/observed by adding an approach connected with swarm algorithms (bees)—HHSABC [29]. In the global optimization, the hybrid turned out to be one of the best hybrids solving this problem. What is worth mentioning is the dynamic form of adapting the values of parameters in the HS, which contributes to the increase of adaptability in generality.

5. Pheromone-Based Harmony Search for ATSP

This section comprises three parts. The first one presents modifications that enable adjustment of the HS to solving instances of the ATSP, the second presents the application of a pheromone in Ant Systems adjusted to searching the route of the salesman, and finally the third one—the structure of the PBHS.

5.1. Harmony Search Adjusted to ATSP

This work was based on a modification of the algorithm suggested in [15]. It assumes representing each pitch by integers, corresponding with the numbers of particular cities, which the salesman is to visit. The order of their occurrence—in a harmony representing the complete route—points towards the sequence of the travel.

While creating a new harmony the sequence of vertices is being considered, which is realized through choosing the next value of pitch based on the generated list of available nodes, occurring in the memorized solution immediately after the last city, which belongs to the result under construction. On the basis of the created structure a city is chosen according to the roulette wheel selection (probability of accepting a given element is dependent on the value of solution objective function, represented by the length of the route, analogically to the approach presented in [30]) or a random non-visited node is drawn (in case when the created list of vertices is empty). As a modification of pitch—connected with the parameter *PAR*—a selection of city (conducted within the area of available nodes), located in the closest proximity of the most recently visited city in the created solution, was adapted.

What was introduced in order to avoid premature convergence is a possibility to reset elements *HM*, when the particular number of iterations *R* is executed, from the last replacing of the

result in the harmony memory. The mechanism assumes storing the best results and drawing the remaining solutions.

5.2. The Role of Pheromone in as Adjusted to Solving TSP

One of the key elements describing the AS is the existence of a pheromone, owing to which the particular agents (the number of which is m) communicate with one another. The route chosen more often by the ant is characterized by a higher intensification of a pheromone, encouraging, as a result, other insects to follow the popular route. The probability of an agent going from the city i to a not visited city j , in time t , is defined on the basis of the following formula:

$$p_{ij}(t) = \frac{\tau_{ij}(t) \cdot \eta_{ij}^\beta}{\sum_{j=1}^n \tau_{ij}(t) \cdot \eta_{ij}^\beta}, \tag{5}$$

where: $\tau_{ij}(t)$ is the size of pheromone trail on the edge connecting the node i and j in time t , η_{ij} is the inverse of the weight of edge between the vertex i and j (short length routes are promoted), n is the number of cities, and parameter β is used to steer the significance of heuristic and pheromone.

Among the various approaches to updating the pheromone, what deserves particular attention is an effective [17] Ant-cycle System, in which the agent k leaves the pheromone after constructing the entire route. $\Delta\tau_{ij}^k(t, t + n)$ adapts the following values:

$$\Delta\tau_{ij}^k(t, t + n) = \begin{cases} \frac{Q_3}{L^k} & \text{if agent } k \text{ went over the edge } (i, j), \\ 0 & \text{otherwise.} \end{cases}, \tag{6}$$

where: Q_3 is a constant (usually taking on value 1 [31]), and L^k —length of the route constructed by an ant k .

After executing n steps the updating of the pheromone trail occurs, according to the following formula:

$$\tau_{ij}(t + n) = (1 - \rho) \cdot \tau_{ij}(t) + \Delta\tau_{ij}(t, t + n), \tag{7}$$

where: $\Delta\tau_{ij}(t, t + n) = \sum_{k=1}^m \Delta\tau_{ij}^k(t, t + n)$ and ρ is parameter representing evaporation of pheromone (value $(1 - \rho) \in [0, 1]$ [31]; used to reduce the impact of historical results and promote the latest one).

The initial size of the pheromone trail τ_0 can be established regardless of the characteristic of a given instance of the problem (e.g., assuming that $\tau_0 = 1$, according to paper [32]) or more sophisticated approaches can be applied, such as e.g., the way suggested in paper [33] based on the Nearest Neighbor Algorithm (NNA). The second of the quoted methods assumes the usage of the following formula:

$$\tau_0 = \frac{1}{n \cdot L_{nn}}, \tag{8}$$

where L_{nn} is the length of the route constructed by the NNA.

5.3. Pheromone-Based Harmony Search for ATSP

In order to increase the effectiveness of the HS, suggested in [15], the process of constructing a new harmony was complemented with the knowledge accumulated by the means of a pheromone (performing the role of long-term memory). To preserve the idea describing the HS, activation of the described mechanism on the basis of HMCR probability was assumed - when the list including successors of the recently visited cities is empty, a selection of the next city, on the basis of pheromone trail, takes place. The updating of the HS global memory is conducted according to Formula (7), after every iteration has been run. Based on the reference books it was assumed that $Q_3 = 1$ [31] and

$\tau_0 = 1$ [32]. The probability of choosing a pitch in position s in harmony H is defined on the basis of a modified Formula (5) (based on the Simple-ACO approach, suggested in [32]), i.e.,

$$p_{ij}(t) = \begin{cases} \frac{\tau_{ij}(t)}{\sum_{j \in A} \tau_{ij}(t)} & \text{if } j \in A, \\ 0 & \text{otherwise.} \end{cases}, \quad (9)$$

where: value i corresponds with the number of a city that is in harmony H in position $s - 1$, and A is a set of cities available for visiting.

The introduction of pheromone memory is connected with the creation of a structure, the size of which is n^2 , and which can influence the problematic implementation of the suggested solution designed for large ATSP instances. In order to eliminate the indicated imperfection it is possible to apply selective pheromone memory, according to the approach suggested in [34]. Pseudocode of the PBHS for ATSP was presented in Algorithm 2.

Algorithm 2 The Pheromone-based Harmony Search for ATSP pseudocode.

```

1: iterations = 0
2: iterationsFromTheLastReplacement = 0
3: Fill the pheromone memory  $\tau$  with the value 1
4: for  $i = 0; i < HMS; i++$  do
5:    $HM[i]$  = stochastically generate feasible solution
6: end for
7: Sort  $HM$ 
8: while iterations <  $IT$  do
9:    $H[0]$  = first city
10:  for  $i = 1; i < n; i++$  do ▷  $n$ —number of cities
11:    Choose random  $r \in (0, 1)$ 
12:    if  $r < HMCR$  then
13:      list = create list containing vertices occurring after  $H[i - 1]$  in  $HM$ 
14:      if list.length > 0 then
15:         $H[i]$  = choose element  $\in$  list according to the roulette wheel
16:      else
17:         $H[i]$  = choose available city based on  $\tau$  and Formula (9)
18:      end if
19:      Choose random  $k \in (0, 1)$ 
20:      if  $k < PAR$  then
21:         $H[i]$  = find nearest and available city from  $H[i - 1]$ 
22:      end if
23:    else
24:       $H[i]$  = choose randomly available city  $\notin H$ 
25:    end if
26:  end for
27:  if  $f(H)$  is better than  $f(HM[HMS - 1])$  then
28:     $HM[HMS - 1] = H$ 
29:    Sort  $HM$ 
30:    iterationsFromTheLastReplacement = 0
31:  else
32:    iterationsFromTheLastReplacement ++
33:  end if
34:  if iterationsFromTheLastReplacement =  $R$  then
35:    for  $i = 1; i < HMS; i++$  do
36:       $HM[i]$  = stochastically generate feasible solution
37:    end for
38:    Sort  $HM$ 
39:    iterationsFromTheLastReplacement = 0
40:  end if
41:  Update  $\tau$  based on Formula (7)
42:  iterations ++
43: end while
44: return  $HM[0]$ 

```

6. Research Methodology

The effectiveness of the suggested approach, taking into consideration the application of a pheromone, was tested on the basis of nineteen instances of the ATSP the characteristic of which was presented in Table 1 (on the basis of [35], where the optimal (best obtainable) route lengths for particular tasks are presented; they are available in the TSPLIB library, described in [36]). In addition, the effectiveness of the methods was verified on the basis of a case study, involving the visit of 40 points located in the Silesian Voivodeship in Poland (it can model e.g., problems related to the distribution of many products; the distance matrix has been prepared with an accuracy to the one meter). A case study map with the locations marked is shown in Figure 1.

Table 1. Description of benchmark tests.

No.	Name	Number of Vertices	Optimum
1	br17	17	39
2	ftv33	34	1286
3	ftv35	36	1473
4	ftv38	39	1530
5	p43	43	5620
6	ftv44	45	1613
7	ftv47	48	1776
8	ry48p	48	14,422
9	ft53	53	6905
10	ftv55	56	1608
11	ftv64	65	1839
12	ft70	70	38,673
13	ftv70	71	1950
14	kro124p	100	36,230
15	ftv170	171	2755
16	rbg323	323	1326
17	rbg358	358	1163
18	rbg403	403	2465
19	rbg443	443	2720

All tasks were solved 30 times by the HS algorithm and three variants of the PBHS (differing in the value of the parameter ρ). To solve the case study we also used the NNA that is often used when companies do not use the right optimization software.

On the basis of work [15] the following parameter values were set: $R = 1000$, $HMS = 5$, $HMCR = 0.98$ and $PAR = 0.25$. Additionally, when analyzing the convergence of the technique (presented in [37]) it was assumed that $IT = 1,000,000$, and as the value of the parameter ρ : 0.25, 0.5 and 0.75 were selected (tests were conducted for each of them in order to select the recommended approach to the designed method).

The algorithms analysed in this work were implemented in language C#, and tests were conducted on a Lenovo Legion Y520 laptop (with: Intel Core i7-7700HQ, 32 GB RAM (SO-DIMM DDR4, 2400 MHz), 1000 GB SATA 7200 RPM, 240 GB SSD M.2 PCIe and Windows 10 Home 64-bit).

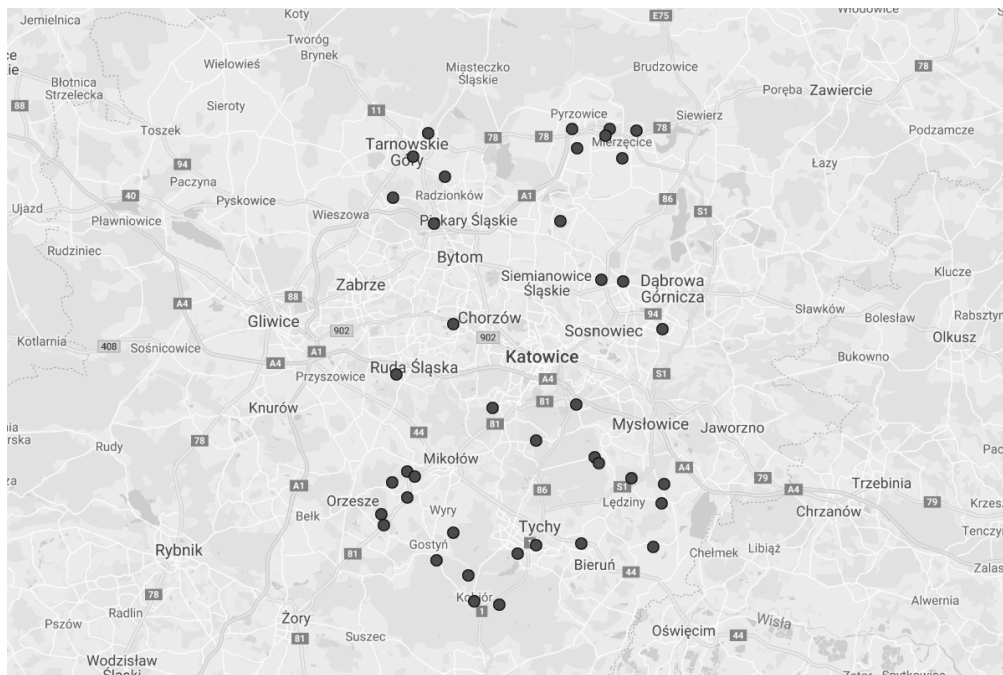


Figure 1. The map for case study with the points to visit.

Both the average error (defined on the basis of Formula (10)) and the actual time of running algorithms were subjected to evaluation. The second measure makes it possible to take into consideration time overhead connected with the introduced modifications and as a result enables defining of the usefulness of the technique in practical adaptations. In addition, we analyzed the number of iterations, after which the algorithm achieved convergence for tasks from TSPLIB.

$$\text{Average error} = \frac{\text{average objective function value} - \text{optimum}}{\text{optimum}} \cdot 100\%. \quad (10)$$

The effectiveness of the proposed method was assessed by comparing the obtained average error with the results achieved by the following algorithms appearing in the literature on the subject:

- Adaptive Multi-Crossover Population Algorithm (AMCPA) [35],
- GA [35],
- Greedy Local Search (GLS) with starting solution constructed by NNA [15],
- Hill Climbing (HC) with starting solution constructed by NNA [15],
- NNA [15].

We chose them, because the first two methods are population-based (like the HS), while the rest of them are classic algorithms commonly used to solve the ATSP instances.

7. Results

This section comprises two parts. The first one presents results for tasks from TSPLIB library, while the second presents the results for the analyzed case study.

7.1. Tasks from TSPLIB

The breakdown of the average error achieved by the particular methods was presented in Table 2 (the best results between the listed methods are in bold). Based on it, it was established that the algorithm devoid of a pheromone defined more favorable results than the PBHS only for two benchmark tests described by the means of a relatively small number of vertices (36 and 45). The effectiveness of the suggested modification was especially emphasized for instances described

by minimum 171 nodes (for task rbg358 limitation of average error by almost 25% was noticed). Regardless of the applied value of parameter ρ , usage of the pheromone substantially increased the effectiveness for the tested method (average error was limited from 13.42 to 10.15, 10.92 and 11.68% for $\rho = 0.25$, $\rho = 0.5$ and $\rho = 0.75$ respectively).

Taking into consideration the total average error and number of tasks, for which the lowest result was observed - among the results defined by the analyzed methods - the most favorable PBHS variant is the technique the parameter value of which is $\rho = 0.25$. The smallest effectiveness is characteristic of the PBHS equipped in a fast pheromone evaporation mechanism ($\rho = 0.75$), which constructed the best route only for the p43 test (difference of the average error between the created solution and results achieved by the remaining methods equals merely 0.01%).

Table 2. Comparison of the average error.

Test Name	Average Error			
	HS	PBHS ($\rho = 0.25$)	PBHS ($\rho = 0.5$)	PBHS ($\rho = 0.75$)
br17	0 ± 0	0 ± 0	0 ± 0	0 ± 0
ftv33	3.63 ± 2.28	1.93 ± 2.09	2.34 ± 2.08	3.41 ± 1.97
ftv35	1.35 ± 0.46	1.43 ± 0.41	1.45 ± 0.39	1.47 ± 0.38
ftv38	1.44 ± 0.71	1.3 ± 0.74	1.35 ± 0.98	1.34 ± 0.57
p43	0.05 ± 0.03	0.05 ± 0.03	0.05 ± 0.02	0.04 ± 0.01
ftv44	1.76 ± 1.73	3.02 ± 1.69	2.12 ± 1.38	2.23 ± 1.48
ftv47	1.95 ± 0.99	2.03 ± 1.09	1.79 ± 0.94	2.08 ± 1.15
ry48p	0.89 ± 0.43	0.77 ± 0.4	1.17 ± 0.91	1.24 ± 0.76
ft53	10.06 ± 3.22	7.97 ± 2.78	10.44 ± 2.99	9.89 ± 3.52
ftv55	2.92 ± 1.72	2.9 ± 1.84	2.8 ± 2.01	2.83 ± 1.26
ftv64	3.09 ± 1.82	2.16 ± 1.41	2.11 ± 1.4	2.42 ± 1.47
ft70	4.35 ± 0.5	3.99 ± 0.43	4.28 ± 0.52	4.06 ± 0.61
ftv70	4.89 ± 1.73	4.62 ± 1.33	4.92 ± 1.41	5.11 ± 1.32
kro124p	8.88 ± 1.62	8.08 ± 1.49	7.6 ± 1.42	8.17 ± 1.64
ftv170	19.15 ± 4.23	15.24 ± 3.35	16.62 ± 4.59	17.58 ± 4.11
rbg323	53.84 ± 2.31	36.89 ± 1.3	40.51 ± 1.9	44.82 ± 2.55
rbg358	77.01 ± 3.86	52.09 ± 2.06	57.41 ± 3.11	62.53 ± 2.56
rbg403	29.27 ± 1.47	23.66 ± 0.88	24.51 ± 0.75	25.66 ± 1.14
rbg443	30.45 ± 1.28	24.73 ± 1.12	26 ± 0.87	27.09 ± 1.12
Average	13.42 ± 1.6	10.15 ± 1.29	10.92 ± 1.46	11.68 ± 1.45

The breakdown of the average error achieved by PBHS and selected methods from the literature on the subject was presented in Table 3 (the best results between the listed methods are in bold) and Figure 2. On their basis, the effectiveness of the proposed approach was found to be significant in relation to the selected techniques described in the literature on the subject. Only for the largest tasks the method obtained a less favorable average error, indicating the need to increase the exploitation process that dominates in GLS and HC.

Table 3. Comparison of the average error obtained by PBHS ($\rho = 0.25$) and selected methods from the literature on the subject.

Test Name	Average Error					
	NNA	GLS	HC	AMCPA	GA	PBHS
br17	135.9%	7.69%	7.69%	0.26%	1.54%	0%
ftv33	30.87%	23.64%	23.64%	7.77%	7.79%	1.93%
ftv35	21.59%	21.38%	21.38%	6.52%	6.2%	1.43%
ftv38	16.21%	10%	10%	5.33%	6.92%	1.3%
p43	2.63%	0.53%	0.96%	0.15%	0.27%	0.05%
ftv44	24.86%	24.18%	24.18%	10.49%	8.38%	3.02%
ftv47	33.67%	28.89%	28.89%	7.21%	4.86%	2.03%
ry48p	16.19%	15.21%	13.81%	2.79%	4.67%	0.77%
ft53	37.78%	30.93%	30.14%	12.64%	11.98%	7.97%
ftv55	25.12%	23.2%	23.2%	11.41%	14.25%	2.9%
ftv64	43.50%	36.54%	36.22%	13.18%	14.86%	2.16%
ft70	11.67%	8.15%	8.98%	4.4%	5.45%	3.99%
ftv70	31.85%	23.85%	23.28%	13.06%	9.97%	4.62%
kro124p	31.12%	26.82%	24.77%	7.67%	10.58%	8.08%
ftv170	42.4%	38.73%	36.88%	46.02%	43.28%	15.24%
rbg323	30.77%	12.37%	12.67%	43.4%	60.11%	36.89%
rbg358	55.8%	17.02%	20.55%	64.05%	74.89%	52.09%
rbg403	43.41%	6.98%	4.87%	17.52%	20.83%	23.66%
rbg443	44.19%	7.21%	7.57%	25.56%	24.29%	24.73%

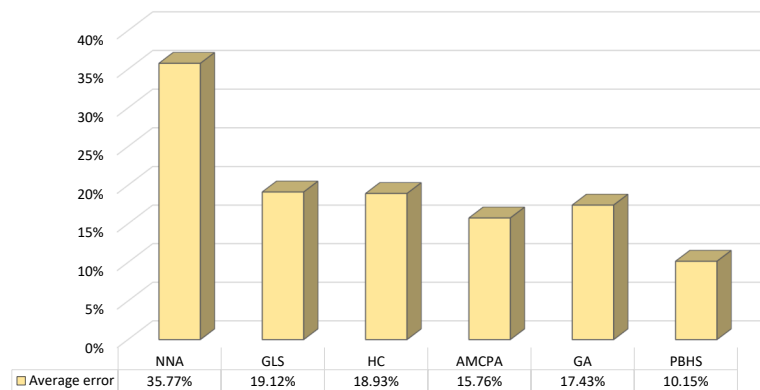


Figure 2. Comparison of the average error obtained by PBHS ($\rho = 0.25$) and selected methods from the literature on the subject.

The breakdown of the running time of particular methods was presented in Table 4. According to the achieved results, the total time of running the method increased by 50% in relation to the technique devoid of the long-term memory. What deserves particular attention is the increase in the time of realizing *IT* iterations by the PBHS for $\rho = 0.25$, in relation to the variants with different values of the indicated parameter. The shortening of the time of running the tasks was due to optimization conducted by the compiler, as a result of which the multiplication operation (of the value $(1 - \rho)$) was substituted by shifting (for $\rho = 0.5$ and $\rho = 0.75$ was 2^{-1} and 2^{-2} achieved). The insignificant differences which appeared when the PBHS was running with $\rho = 0.5$ and $\rho = 0.75$ were probably caused by an unequal processor overload.

Table 4. Comparison of the execution time.

Test Name	Average Execution Time [s]			
	HS	PBHS ($\rho = 0.25$)	PBHS ($\rho = 0.5$)	PBHS ($\rho = 0.75$)
br17	22.16 ± 0.45	38.32 ± 1.26	24.96 ± 0.64	29.66 ± 1.28
ftv33	46.11 ± 1.25	112.78 ± 5.17	55.51 ± 1.2	68.61 ± 1.54
ftv35	48.96 ± 0.73	125.04 ± 3.05	60.07 ± 0.81	74.12 ± 1.07
ftv38	53.83 ± 0.67	146.37 ± 2.7	63.5 ± 3.62	84.72 ± 7.75
p43	60.14 ± 1.76	161.71 ± 8.49	69.61 ± 3.03	90.46 ± 2.12
ftv44	63.58 ± 1.21	196.78 ± 12.35	79.07 ± 2.82	100.16 ± 3.74
ftv47	68.39 ± 0.67	209.05 ± 4.53	84.37 ± 1.31	103.01 ± 4.28
ry48p	68.15 ± 0.97	174.53 ± 22.27	84.15 ± 1.24	103.96 ± 1.88
ft53	76.53 ± 0.81	216.73 ± 14.73	99.08 ± 2.79	119.61 ± 2.34
ftv55	87.16 ± 2.51	269.93 ± 8	106.7 ± 4.95	122.15 ± 8.71
ftv64	106 ± 3.3	372.46 ± 4.6	136.9 ± 3.25	163.65 ± 6.82
ft70	109.36 ± 4.03	426.12 ± 11.31	149.33 ± 11.68	169.16 ± 7.6
ftv70	116.43 ± 3.85	421.5 ± 8.22	175.58 ± 3.88	178.33 ± 8.05
kro124p	205.5 ± 5.3	803.88 ± 41.47	296.29 ± 26.43	308.59 ± 20.09
ftv170	418.36 ± 23.65	2077.29 ± 165.99	610.74 ± 39.72	646.33 ± 22.48
rbg323	1033.5 ± 56.87	6140.01 ± 405.66	1513.5 ± 261.93	1640.2 ± 130.42
rbg358	1170.66 ± 135.41	7191.41 ± 403.9	2069.71 ± 50.73	1919.57 ± 129.31
rbg403	1515.98 ± 35.33	9266.52 ± 237.28	2355.65 ± 211.82	1964.43 ± 122.16
rbg443	1665 ± 130.83	11136.23 ± 275	2371.31 ± 156.91	2273.62 ± 159.5
Average	365.04 ± 21.56	2078.25 ± 86.1	547.69 ± 41.51	534.76 ± 33.75

The number of iterations after which different PBHS variants have reached convergence is shown in Figure 3 (using the popular box plot, described e.g., in the paper [38]). On their basis, there were no significant differences between the individual PBHS settings to achieve convergence. In addition, the scatter of iterations indicates the choice of the not too low value of *IT* for the tested set (only for the smallest task br17 it could be reduced without any loss of efficiency).

The average error—defined by the particular methods—was subjected to the Wilcoxon Signed-Rank Test, with the application of the R package and the wilcox.test function (applying the following values of parameters: paired = TRUE, alternative = “less”, exact = F, correct = F). The following pairs of methods were used in the process (marked as *O1* and *O2*), and the assumed test significance level value was 0.05 (achieved *p*-values, which were characterized by a lower result, were distinguished by bold and they cause to accept the alternative hypothesis, according to which *O1* achieved lower results than *O2*). The results of the test were presented in Table 5. On their basis, it was established that taking into consideration only the value of the objective function of the solution, it is recommended to use the PBHS with $\rho = 0.25$. Additionally, no grounds were found to negate the hypothesis which claims that the usage of a pheromone makes it possible to achieve better results than the ones defined by the HS without global memory.

Table 5. Wilcoxon Signed-Rank Test results for the value of the objective function.

<i>O1/O2</i>	HS	$\rho = 0.25$	$\rho = 0.5$	$\rho = 0.75$
HS	N/A	1	1	1
$\rho = 0.25$	1.29941×10^{-46}	N/A	1.06687×10^{-25}	6.89734×10^{-42}
$\rho = 0.5$	1.61089×10^{-36}	1	N/A	4.60633×10^{-31}
$\rho = 0.75$	1.78842×10^{-30}	1	1	N/A

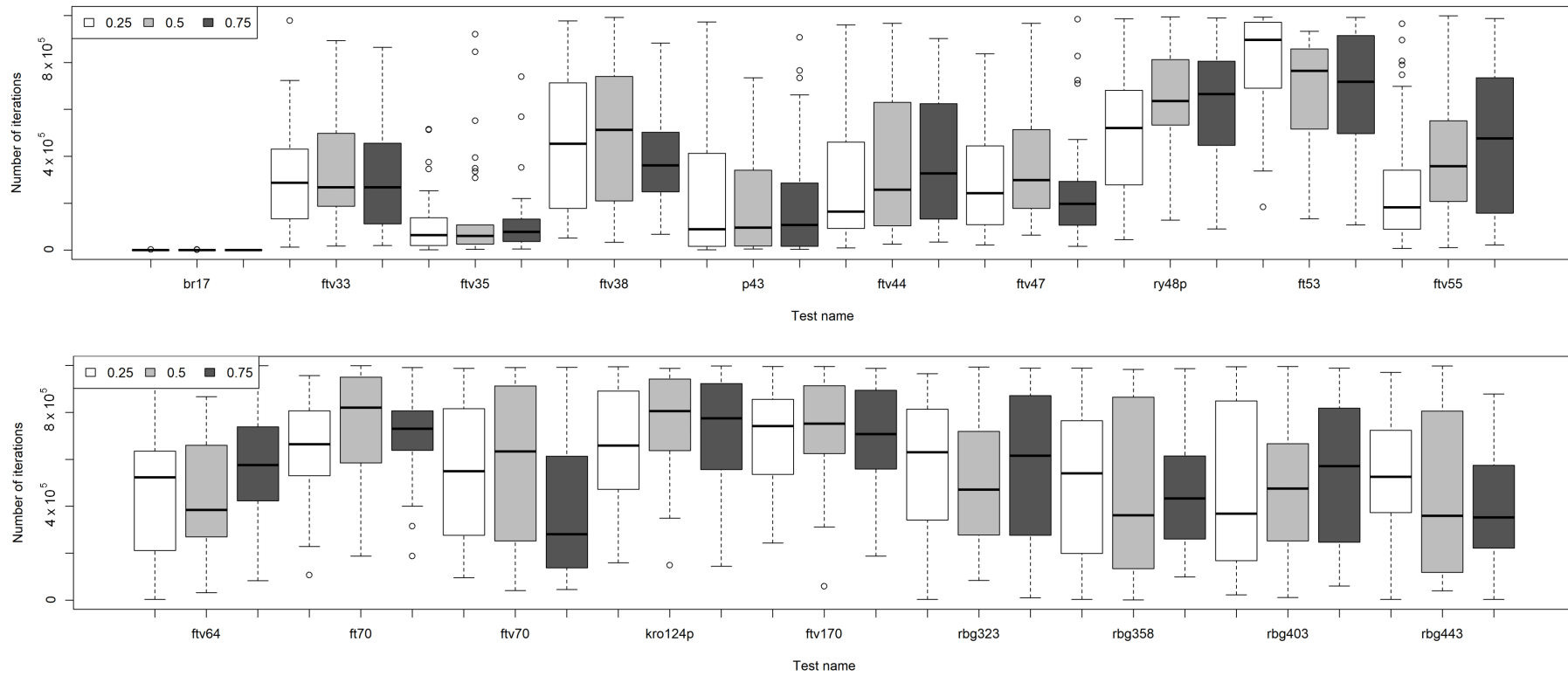


Figure 3. PBHS convergence.

7.2. Case Study

Figure 4 presents a comparison of results (total distance) obtained for a case study by PBHS, HS and NNA (starting to create a solution from node number 1). Based on them, the effectiveness of the developed PBHS was found - it allowed to shorten the average length of the route from 278.12 (HS results) to 276.67 ($\rho = 0.25$), 276.05 ($\rho = 0.5$) and 275.88 ($\rho = 0.75$) km. It is also worth emphasizing the higher predictability of PBHS results (smaller spread) than HS results.

Limiting the average length of the route by 21.58 km, in relation to the result constructed by NNA, indicates significant benefits that may arise as a result of the use of PBHS in business practice. The relatively small reduction in the length of the route by PBHS compared to the HS result may allow for a significant reduction of costs when frequent route optimization is required.

In contrast to previous results, the recommended ρ value for the case study is 0.75, however, the slight difference between the length of the obtained routes indicates the possibility of the influence of nondeterminism on the results.

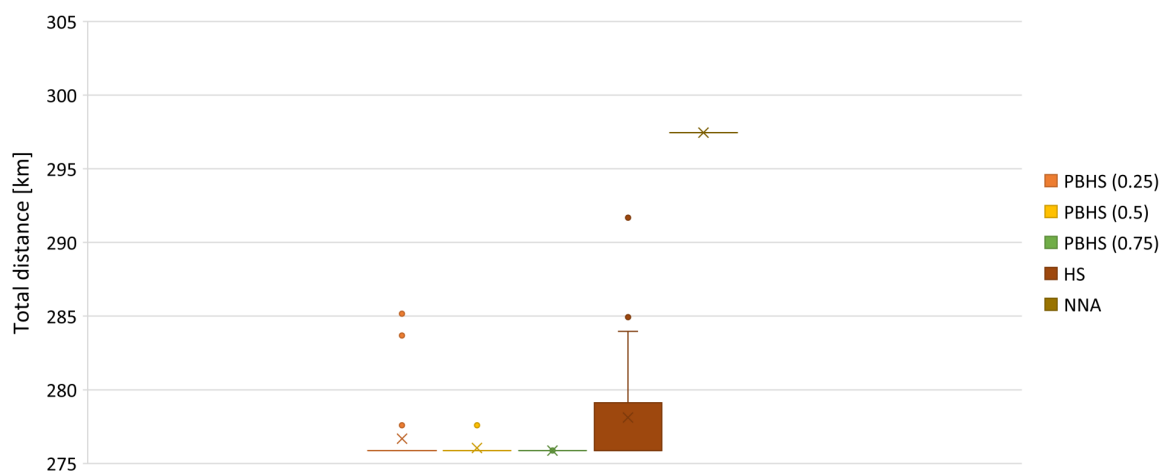


Figure 4. Comparison of results for case study (total distance).

Figure 5 presents a comparison of results (execution time) obtained for case study by PBHS and HS. The results obtained confirmed the conclusions resulting from the analysis of a set of tasks from the TSPLIB library, however, for most practical applications, the execution time of PBHS with $\rho = 0.25$ (characterized by the longest time), amounting to about three minutes is acceptable, allowing its use.

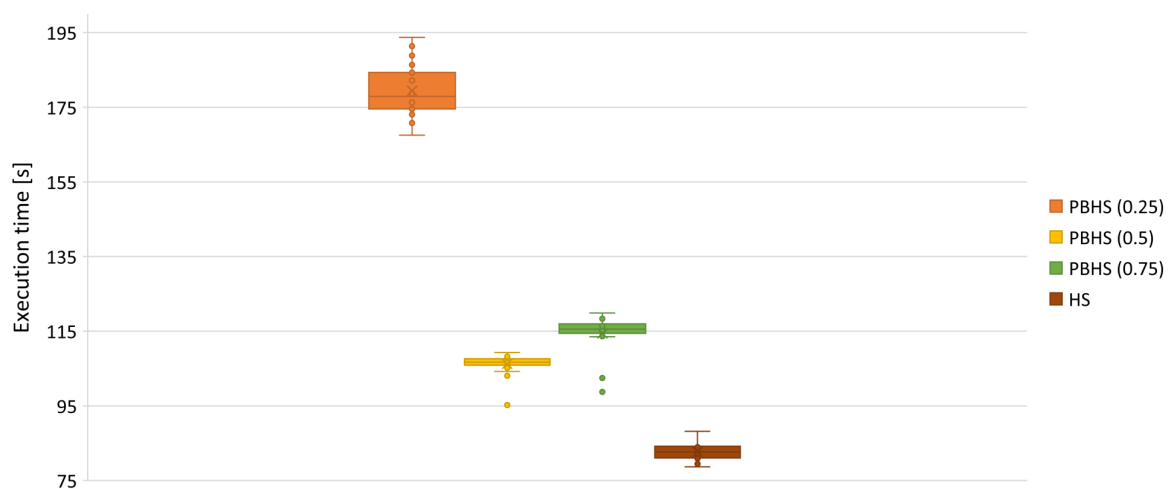


Figure 5. Comparison of results for case study (execution time).

8. Conclusions and Future Work

As a result of the conducted research the effectiveness of the application of pheromone in HS adapted to solving the ATSP was confirmed. The combination of the advantages of the algorithm proposed in [15] with the version of the Simple-ACO (suggested in [32]), allowed to reduce the average error from 13.42 to 10.15%.

A comparison of the results obtained with the results of selected methods from the subject literature indicates significant effectiveness of the proposed approach, allowing it to be used in business practice. In addition, the presented case study showed significant advantages of the method and verified its suitability both in terms of execution time and the possibility of optimizing the route.

For objectives requiring the relatively fast achievement of a result setting the value of the ρ parameter in PBHS at 0.5 is recommended. In the remaining cases, the application of $\rho = 0.25$ is recommended.

Further work concerning PBHS can concentrate on a more detailed adjustment of the value of parameter ρ and conducting the hybridization of the technique with different methods, aimed at increasing the effectiveness of the suggested approach. Additionally, what is recommended is to prepare a parallel version of the PBHS (e.g., based on the approach presented in [39]) and adjusting the algorithm to the remaining combinatorial optimization problems. It is also worth checking the efficiency of the algorithm for another τ_0 value (e.g., using the approach proposed in [33]). Comprehensive studies should also be performed to compare the effectiveness of the proposed PBHS and other HS variants, including Improved HS [40], GHS [28] and Self-adaptive GHS [41].

Author Contributions: Conceptualization, K.S. and U.B.; methodology, K.S. and U.B.; software, K.S.; validation, K.S. and U.B.; formal analysis, K.S. and U.B.; investigation, K.S. and U.B.; resources, K.S. and U.B.; data curation, K.S. and U.B.; writing—original draft preparation, K.S. and U.B.; visualization, K.S. and U.B.; supervision, U.B.; project administration, K.S. and U.B.; funding acquisition, U.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

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

Abbreviations

ACO	Ant Colony Optimization
AMCPA	Adaptive Multi-Crossover Population Algorithm
AS	Ant Systems
ATSP	Asymmetric Traveling Salesman Problem
CS	Cuckoo Search
GA	Genetic Algorithm
GHS	Global-best Harmony Search
GLS	Greedy Local Search
HC	Hill Climbing
HS	Harmony Search
MTZ	Miller, Tucker and Zemlin
NN	Neural Networks
NNA	Nearest Neighbor Algorithm
PBHS	Pheromone-based Harmony Search
PSO	Particle Swarm Optimization
TSP	Traveling Salesman Problem

References

1. De, A.; Choudhary, A.; Tiwari, M.K. Multiobjective Approach for Sustainable Ship Routing and Scheduling With Draft Restrictions. *IEEE Trans. Eng. Manag.* **2019**, *66*, 35–51. [[CrossRef](#)]
2. Borovinšek, M.; Ekren, B.; Burinskienė, A.; Lerher, T. Multi-objective optimisation model of shuttle-based storage and retrieval system. *Transport* **2017**, *32*, 120–137. [[CrossRef](#)]

3. De, A.; Mamanduru, V.K.R.; Gunasekaran, A.; Subramanian, N.; Tiwari, M.K. Composite particle algorithm for sustainable integrated dynamic ship routing and scheduling optimization. *Comput. Ind. Eng.* **2016**, *96*, 201–215. [[CrossRef](#)]
4. De, A.; Kumar, S.K.; Gunasekaran, A.; Tiwari, M.K. Sustainable maritime inventory routing problem with time window constraints. *Eng. Appl. Artif. Intell.* **2017**, *61*, 77–95. [[CrossRef](#)]
5. Kim, J.H. Harmony Search Algorithm: A Unique Music-inspired Algorithm. *Procedia Eng.* **2016**, *154*, 1401–1405. [[CrossRef](#)]
6. Weyland, D. A Rigorous Analysis of the Harmony Search Algorithm: How the Research Community Can Be Misled by a “Novel” Methodology. *Int. J. Appl. Metaheuristic Comput.* **2010**, *1*, 50–60. [[CrossRef](#)]
7. Bashiri-Atrabi, H.; Qaderi, K.; Rheinheimer, D.E.; Sharifi, E. Application of Harmony Search Algorithm to Reservoir Operation Optimization. *Water Resour. Manag.* **2015**, *29*, 5729–5748. [[CrossRef](#)]
8. Bo, G.; Huang, M.; Ip, W.H.; Wang, X. The Application of Harmony Search in Fourth-Party Logistics Routing Problems. In *Recent Advances in Harmony Search Algorithm*; Geem, Z.W., Ed.; Springer: Berlin/Heidelberg, Germany, 2010; pp. 135–145. [[CrossRef](#)]
9. Ayachi, I.; Kammarti, R.; Ksouri, M.; Borne, P. Harmony search to solve the container storage problem with different container types. *arXiv* **2013**, arXiv:abs/1305.7254.
10. Syberfeldt, A.; Rogstrom, J.; Geertsen, A. Simulation-based Optimization of a Real-world Travelling Salesman Problem Using an Evolutionary Algorithm with a Repair Function. *Int. J. Artif. Intell. Expert Syst. IJAE* **2015**, *6*, 27–39.
11. Nowakowski, P.; Szwarc, K.; Boryczka, U. Vehicle route planning in e-waste mobile collection on demand supported by artificial intelligence algorithms. *Transp. Res. Part Transp. Environ.* **2018**, *63*, 1–22. [[CrossRef](#)]
12. Srivastava, S.K. Network design for reverse logistics. *Omega* **2008**, *36*, 535–548. [[CrossRef](#)]
13. Kalafarski, M.; Czech, P.; Mrówczyńska, B.; Sierpiński, G.; Turoń, K. Przykład zastosowania algorytmu mrówkowego do usprawnienia procesu dystrybucji produktów szybko psujących. *Autobusy Tech. Eksploat. Syst. Transp.* **2018**, *19*, 1051–1058.
14. Pongchairerks, P.; Putponpai, J. Deterministic, stochastic and local search algorithms for real-world ATSP instances. *Int. J. Logist. Syst. Manag.* **2016**, *24*, 265–282. [[CrossRef](#)]
15. Boryczka, U.; Szwarc, K. The Adaptation of the Harmony Search Algorithm to the ATSP. In *Intelligent Information and Database Systems*; Nguyen, N.T., Hoang, D.H., Hong, T.P., Pham, H., Trawiński, B., Eds.; Springer International Publishing: Cham, Germany, 2018; pp. 341–351. [[CrossRef](#)]
16. Boryczka, U.; Skinderowicz, R.; Świstowski, D. Comparative study: ACO and EC for TSP. *Czas. Pr. Nauk. Politech. Warsz.* **2006**, *156*, 69–77.
17. Colorni, A.; Dorigo, M.; Maniezzo, V. Distributed Optimization by Ant Colonies. In *European Conference on Artificial Life*; Elsevier Publishing: Paris, France, 1991; pp. 134–142.
18. Szwarc, K.; Boryczka, U. A Comparative Study of Techniques for Avoiding Premature Convergence in Harmony Search Algorithm. In *Intelligent Information and Database Systems*; Nguyen, N.T., Gaol, F.L., Hong, T.P., Trawiński, B., Eds.; Springer International Publishing: Cham, Switzerland, 2019; pp. 203–214. [[CrossRef](#)]
19. Yang, X.S. Harmony Search as a Metaheuristic Algorithm. In *Music-Inspired Harmony Search Algorithm: Theory and Applications*; Geem, Z.W., Ed.; Springer: Berlin/Heidelberg, Germany, 2009; pp. 1–14. [[CrossRef](#)]
20. Geem, Z.W.; Kim, J.H.; Loganathan, G. A New Heuristic Optimization Algorithm: Harmony Search. *Simulation* **2001**, *76*, 60–68. [[CrossRef](#)]
21. Öncan, T.; Altinel, K.; Laporte, G. A comparative analysis of several asymmetric traveling salesman problem formulations. *Comput. Oper. Res.* **2009**, *36*, 637–654. [[CrossRef](#)]
22. Wolpert, D.H.; Macready, W.G. No free lunch theorems for optimization. *IEEE Trans. Evol. Comput.* **1997**, *1*, 67–82. [[CrossRef](#)]
23. Sinsuphan, N.; Leeton, U.; Kulworawanichpong, T. Optimal Power Flow Solution Using Improved Harmony Search Method. *Appl. Soft Comput.* **2013**, *13*, 2364–2374. [[CrossRef](#)]
24. Lee, J.H.; Yoon, Y.S. Modified Harmony Search Algorithm and Neural Networks for Concrete Mix Proportion Design. *J. Comput. Civ. Eng.* **2009**, *23*, 57–61. [[CrossRef](#)]
25. Feng, Y.; Wang, G.G.; Gao, X.Z. A Novel Hybrid Cuckoo Search Algorithm with Global Harmony Search for 0-1 Knapsack Problems. *Int. J. Comput. Intell. Syst.* **2016**, *9*, 1174–1190. [[CrossRef](#)]
26. Younes, M.; Khodja, F. A Hybrid Harmony Search Algorithm Approach for Optimal Power Flow. *Leonardo J. Sci.* **2012**, *20*, 125–144.

27. Kaveh, A.; Talatahari, S. Particle Swarm Optimizer, Ant Colony Strategy and Harmony Search Scheme Hybridized for Optimization of Truss Structures. *Comput. Struct.* **2009**, *87*, 267–283. [[CrossRef](#)]
28. Omran, M.G.H.; Mahdavi, M. Global-best harmony search. *Appl. Math. Comput.* **2008**, *198*, 643–656. [[CrossRef](#)]
29. Wu, B.; Qian, C.; Ni, W.; Fan, S. Hybrid harmony search and artificial bee colony algorithm for global optimization problems. *Comput. Math. Appl.* **2012**, *64*, 2621–2634. [[CrossRef](#)]
30. Komaki, M.; Sheikh, S.; Teymourian, E. A Hybrid Harmony Search algorithm to minimize total weighted tardiness in the permutation flow shop. In Proceedings of the 2014 IEEE Symposium on Computational Intelligence in Production and Logistics Systems (CIPLS), Orlando, FL, USA, 9–12 December 2014; pp. 1–8. [[CrossRef](#)]
31. Boryczka, U. *Algorytmy Optymalizacji Mrowiskowej*; Prace Naukowe Uniwersytetu Śląskiego w Katowicach, Wydawnictwo Uniwersytetu Śląskiego: Katowice, Poland, 2006.
32. Dorigo, M.; Stützle, T. *An Experimental Study of the Simple Ant Colony Optimization Algorithm*; The MIT Press: London, England, 2001; pp. 253–258.
33. Corne, D.; Dorigo, M.; Glover, F.; Dasgupta, D.; Moscato, P.; Poli, R.; Price, K.V. (Eds.) *New Ideas in Optimization*; McGraw-Hill Ltd.: Maidenhead, UK, 1999.
34. Skinderowicz, R. Ant Colony System with Selective Pheromone Memory for TSP. ICCCI (2). In *Lecture Notes in Computer Science*; Springer: Cham, Switzerland, 2012; Volume 7654, pp. 483–492.
35. Osaba, E.; Díaz, F.; Onieva, E.; Carballedo, R.; Perallos, A. AMCPA: A Population Metaheuristic with Adaptive Crossover Probability and Multi-Crossover Mechanism for Solving Combinatorial Optimization Problems. *Int. J. Artif. Intell.* **2014**, *12*, 1–23.
36. Reinelt, G. TSPLIB—A Traveling Salesman Problem Library. *Orsa J. Comput.* **1991**, *3*, 376–384. [[CrossRef](#)]
37. Boryczka, U.; Szwarc, K. The adaptation of the harmony search algorithm to the ATSP with the evaluation of the influence of the pitch adjustment place on the quality of results. *J. Inf. Telecommun.* **2019**, *3*, 2–18. [[CrossRef](#)]
38. Williamson, D.F.; Parker, R.A.; Kendrick, J.S. The Box Plot: A Simple Visual Method to Interpret Data. *Ann. Intern. Med.* **1989**, *110*, 916–921. [[CrossRef](#)]
39. Szwarc, K.; Boryczka, U. Analysis of Different Approaches to Designing the Parallel Harmony Search Algorithm for ATSP. In *Intelligent Information and Database Systems*; Nguyen, N.T., Gaol, F.L., Hong, T.P., Trawiński, B., Eds.; Springer International Publishing: Cham, The Netherlands, 2019; pp. 215–227.
40. Mahdavi, M.; Fesanghary, M.; Damangir, E. An improved harmony search algorithm for solving optimization problems. *Appl. Math. Comput.* **2007**, *188*, 1567–1579. [[CrossRef](#)]
41. Pan, Q.K.; Suganthan, P.; Tasgetiren, M.F.; Liang, J. A self-adaptive global best harmony search algorithm for continuous optimization problems. *Appl. Math. Comput.* **2010**, *216*, 830–848. [[CrossRef](#)]



© 2020 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 (<http://creativecommons.org/licenses/by/4.0/>).