ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

A MULTI-POPULATION HARMONY SEARCH ALGORITHM FOR THE DYNAMIC TRAVELLING SALESMAN PROBLEM WITH TRAFFIC FACTORS

[1 2]MOHANAD MUAYAD JOHN JURJEE, [3]HAFIZ MOHD SARIM, [4]NOORA HANI ABDULMAJEED AL-DABBAGH, [5]ERNA BUDHIARTI NABABAN

[1 4]Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia, Bangi, Malaysia

[2]College of Science, Al-Mustansiriyah University, Baghdad, Iraq

[3]Data Mining and Optimization Research Group, Centre for Artificial Intelligence Technology, Universiti Kebangsaan Malaysia, Bangi, Malaysia

[5]Faculty of Computer Science and Information Technology, Universitas Sumatera Utara

E-mail: [1 2]muhannad_manchester@yahoo.com, [3]hms@ukm.edu.my, [4]noora.aldabbagh88@gmail.com, [5]ernabrn@usu.ac.id

ABSTRACT

Recently, there has been a growing attention to employ evolutionary algorithms (EAs) in addressing dynamic optimisation problems (DOPs) due to its significance in real world applications. The most notable challenge when solving DOPs is that the objective should not only attempt to seek the global optimum by an efficient way, but be able to keep track the optimal solution during the environmental changes. Thus, several mechanisms have been developed for EAs in order to improve the search performance of the algorithm in accommodating the dynamic changes such as by increasing the diversity of the population. Among these strategies, the multi-population mechanism has been found beneficial for EAs for DOPs. Dynamic travelling salesman problems (DTSPs) are categorised under DOPs. In the Travelling Salesman Problem (TSP), a salesman wants to distribute items sold in different cities starting from his home city and returning after he visited all the cities to his starting city again by optimising his time and tour efficiently. However, in the DTSPs, it is more challenging to consider the traffic delays that may affect the route of the salesman and change the time planned beforehand. Therefore, the salesman will optimise his time again and find a new alternative route to avoid long traffic delays. The presented work aims to build upon the state of the art research methodologies for the DTSPs with traffic factors, where in order to cope with the dynamic behaviour, a multi-population approach is applied to harmony search algorithm that mimics the musical process of trying to find a state of harmony. Moreover, a multiple pitch adjustment rate (PAR) strategy is proposed since PAR assumed to be the moving rate from one city to the nearest city in the TSP. The performance of the proposed multi-population HS algorithm is verified on two variations of DTSPs with traffic factors, i.e., random and cyclic traffic delays. Based on different DTSP test cases, the experimental results show that the proposed approach is able to obtain competitive results when compared to the best-known results in the scientific literature.

Keywords: *Dynamic Optimisation, Travelling Salesman Problem (TSP), Harmony Search, Multi-Population Approach, Diversity*

1. INTRODUCTION

Problem optimisation in dynamic environments has attracted an increasing attention because of its significance in real world problems [1]. In the dynamic optimisation problems (DOPs), there is a probability to change the problem through the execution of an algorithm [2]. These changes are touching the objective function, problem instance, and decision variables. Therefore, it may move the optimum. So that, the problem will be more realistic, challenging and pragmatic.

However, when solving the static optimisation problems (SOPs), the objective is to get the optimal solution by an efficient way. But the objective, when solving the DOPs, is about tracking an optimal solution during the environmental changes [3].

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

The travelling salesman problem (TSP) is one of the major combinatorial optimisation problems (COPs) and of the most intensively researched problems in computational mathematics. The TSP belongs to the class of NP-complete problems since it requires exponential time to create the optimal solution [4]. It has excited studies by researchers of varied fields, e.g., computer science, mathematics, chemistry, physics and psychology. Teachers also involve the TSP to make discrete mathematics known by students in schools or universities.

In the dynamic travelling salesman problem (DTSP), cities may be added or deleted and the distances between each pair of cities may be changed during the tour. However, the problem will be more realistic if it is subjected to consider traffic jams which may affect the tour and change the original time. In this case, the salesman must optimise the time again and find his alternative path to avoid the traffic jams. Recently, several methods have been used to solve various types of DTSPs due to its significance for a large number of real-world implementations [5].

Over the years, population based methods have been applied successfully for the DOPs [6]. These methods are dealing with a population of solutions that are dispersed over the entire search space, and that will help the algorithm to track the environmental changes since each solution in the population is assigned to a various region in the search space [1]. However, it is not possible for the population based methods that were improved to handle SOPs, to solve DOPs easily.

It is evident that to deal with dynamic optimisation, the proposed population-based methods have to combine with some mechanisms that would adapt their conduct by modifying their components to be suitable for accommodating the dynamic changes. Increasing diversity after a dynamic change is a simple mechanism to modify the behavior of the population-based methods in order to cope with dynamic environments [7]. Immigrants schemes, memory approaches and the multi-population mechanism have been found valuable when applied to evolutionary algorithms for solving DOPs [8]. Due to the developing strategies that managed to maintain the diversity, some population-based algorithms have been used successfully to solve DOPs, e.g., Genetic Algorithms (GA) [9], Ant Colony Optimisation (ACO) algorithm [10], Particle Swarm Optimisation (PSO) algorithm [11] and Harmony Search (HS) algorithm [12].

DYNAMIC TRAVELLING SALESMAN PROBLEMS (DTSPS)

In the DTSP with random traffic delays, a random number that represented the traffic delays will be generated during the running of an algorithm, where the number of traffic delays is greater or equal to the lower bound of the traffic factor and smaller or equal to its upper bound. The visited environments are not guaranteed to appear once more in the random pattern.

But with the cyclic pattern, the visited environments are guaranteed to appear another time. The cyclic variation of DTSP is showing a practical idea more than random DTSPs because a 24-hours traffic jam in the day is considered.

The cyclic environment can be structured by creating various dynamic states with traffic factors as the base states. The DTSP environments with either lower, average, or higher traffic are represented by these dynamic cases. Therefore, the environment will be cyclic in a fixed logical circle within different dynamic cases.

Environments with various traffic factors can be created depending on the time of the day. A higher expectation is given to generate a random number that represented the traffic delays, closer to the upper bound of traffic factor t_{ij} in the case of rush-hour periods when traffic is at its heaviest for example. In the opposite way, a higher expectation is given to generate the random number of traffic delays, closer to the lower bound of traffic factor t_{ij} in the case of evening-hour periods.

3. THE MULTI-POPULATION HARMONY SEARCH ALGORITHM

This work aims to investigate the performance of the suggested HS algorithm to solve the DTSP with traffic factors. However, the application of population-based algorithms for the DOPs is impractical. To deal with DOPs, the proposed population-based method has to combine with a mechanism that would adapt its conduct by modifying its components to be suitable to adjust the dynamic changes. In this work, HS algorithm is hybridised with multi-population approach to maintain the population diversity.

3.1. Harmony Search

Harmony Search (HS) is a search heuristic that mimics the music improvisation of different jazz musicians where they are trying to find better harmonies and attempt to adjust the pitches of their instruments [13], and so, all harmonies are taking into account in jazz music and optimised due to

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

underlying objectives. The musicians are starting with some harmonies and then try to achieve the better harmony through the improvisation process by refining the current harmony or producing a new one. Then, the last harmony will be evaluated in order to accept it or discard it. This analogy used by the researchers to derive the population-based search heuristic that employed for solving optimisation problems where the musicians correspond to decision variables and the harmonies are identified with the solutions.

HS has been subject of a lot of publications since its introduction and has been applied in many fields such as Sudoku puzzle solving, Web page clustering, Water network design, Dam scheduling, Soil stability analysis, Energy system dispatch and Ecological conservation [14].

HS consists of five steps in order to imitate the music improvisation process. These steps are:
Step 1: Initialise the problem and algorithm parameters,
Step 2: Initialise the harmony memory,
Step 3: Improvise a new harmony,
Step 4: Update the harmony memory, and
Step 5: Termination criterion.

3.2. Hybridised HS with Multi-Population Approach

The population diversity mechanism that embedded within HS algorithm in order to maintain the population diversity, is presented in this section. In this work, the multi-population mechanism divides the population into two sub-populations, i.e. the quality population and the diversity population. Then the solutions in each sub-population will be ordered; in the quality population to the objective function and the diversity population to the dissimilarity value.

The proposed multi-population HS algorithm needs to keep track of the changes during the search process by maintaining the population diversity in order to cope with the dynamic changes.

After the harmony memory (HM) is initialised in step 2, the initial HM will divided into two sub-populations with the same size for both at the first iteration. The 10 top quality solutions in the initial HM will be selected to the quality harmony memory (QHM) and the 10 most diverse solutions that remain in the HM will be selected to the diversity harmony memory (DHM). The solutions in the QHM will be ordered to their objective function and the solutions in the DHM will be ordered to their dissimilarity value.

In the third step of the proposed algorithm, two subsets will be derived from the HM, i.e. QHM and DHM, as follows:

i. Select the top 10 quality solutions from the HM and store them in the QHM.

ii. The diversity of the remaining solutions in the initial HM will be measured.

iii. Select the 10 solutions from HM that are least similar to the solutions in the QHM, to be stored in the DHM [15].

After a new harmony is improvised, the decision variable value will be selected from QHM solutions before using the pitch adjustment rule if the random number is less than HMCR. Otherwise, it will be selected from DHM solutions.

If the quality of the generated harmony vector is better than the worst quality of harmony vector stored in QHM, then the worst harmony vector in QHM is replaced by the new vector, else if the generated vector is more diverse than the less diverse vector in DHM, then the less diverse vector in DHM is replaced by the new vector. Otherwise, the new vector is ignored.

The hybrid HS algorithm with multi-population approach that implemented and applied to solve the DTSP with traffic factors, is shown in Fig. 1.

3.3. The Multiple PAR Strategy

In this work, the multiple PAR strategy is proposed for the multi-population HS algorithm. Pitch Adjustment Rate (PAR) is the moving rate from one value in the harmony memory to a neighboring value. In this work, three PARs are used, i.e., the rates of moving to nearest city = 0.35, second nearest city = 0.105 and third nearest city = 0.045 [16], since the PAR becomes the moving rate from one city to the nearest city in the TSP.

4. EXPERIMENTAL SETUP

Several experiments are carried out in this work in order to investigate the effect of the multi-population approach and multiple-PAR strategy on HS algorithm for the DTSP.

The proposed method was tested on the DTSP instances that are generated from three stationary benchmark TSP instances that can be freely downloaded from TSPLIB, i.e., kroA100, kroA150, and kroA200, which represent small, medium, and larger scale problem instances in the related works, respectively [5].

Also, the performance of the proposed multi-population HS algorithm for the DTSP is compared with the performance of the basic HS algorithm. In addition to the three datasets that are mentioned above, four datasets taken from TSPLIB are used in

Journal of Theoretical and Applied Information Technology
31st January 2017. Vol.95. No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

order to examine the performance of the suggested algorithm against the standard HS algorithm, i.e., eil51, lin318, pr439 and rat783.

In this work, two kinds of DTSPs are generated, i.e., random and cyclic traffic factors, with lower bound of the potential traffic jams (F_L) = 1 and upper bound of the potential traffics (F_U) = 5. Then, the values of the random number (*R*) that represents the traffic jams ∈ [1, 5]. In addition, three cyclic states are used in cyclic DTSPs.

For both random and cyclic types, the frequency parameter (*f*) value for the environmental changes was set to 5 and 100 in order to indicate fast and slow dynamic changes, respectively. Also, the value of the magnitude parameter of the dynamic changes (*m*) was set to 0.1, 0.25, 0.5, and 0.75 in order to indicate the degree of dynamic changes from small, to medium, and large, respectively [5].

Consequently, eight dynamic test DTSP instances are generated from each dataset, i.e. four values of *m* × two values of *f*. Therefore, 24 dynamic test cases are used to analyse the adaptation capabilities of the proposed algorithm on the DTSP with traffic factors for each type of DTSPs, i.e., three TSP instances × eight cases each.

In order to analyse the adaptation capabilities of the multi-population HS algorithm on the DTSP against the basic HS algorithm, 56 dynamic test cases are used for both of random and cyclic types, i.e., seven TSP instances × eight cases each.

For the suggested multi-population HS algorithm on a DTSP instance, *N* = 30 runs are executed on the same environmental changes. The algorithm are executed for *G* = 1000 iterations.

5. COMPARISON BETWEEN THE MULTI-POPULATION HS ALGORITHM WITH THE BASIC HS ALGORITHM

In this section, the results obtained by the multi-population HS algorithm using multiple PARs strategy (denoted MPHSA-3PARs) are compared against the basic HS algorithm (denoted HSA) and also the multi-population HS algorithm using one PAR (denoted MPHSA-1PAR).

The basic experimental results in DTSP with random traffic factors in case of fast environmental changes (f = 5) are given in Table 1, and in case of slow environmental changes (f = 100) are given in Table 2. The basic experimental results in DTSP with cyclic traffic factors in case of fast environmental changes (f = 5) are given in Table 3, and in case of slow environmental changes (f = 100) are given in Table 4. The best results are presented in bold.

Statistical test results regarding the offline performance of HSA, MPHSA-1PAR and MPHSA-3PARs on random and cyclic DTSPs that tested on kroA100, kroA150 and kroA200 instances are given in Table 5, and that tested on eil51, lin318, pr439 and rat783 instances are given in Table 6.

The results show that the multi-population HS algorithm using multiple PARs strategy is able to obtain better results than the standard HSA and the multi-population HS algorithm using one PAR in all of the seven tested datasets with random and cyclic traffic factors, and in all of the different cases.

In order to measure the solution diversity, a frequency matrix is used to store the frequency of assigning a city to the same slot. Thus, the frequency matrix saves how many times a salesman has been assigned to the same path. The frequency matrix is initialised to zero and updated if any solution is improved by the algorithm. An example of a solution and its identical frequency matrix is shown in Fig. 2 where the rows represent the cities and the columns represent slots (the available locations). For example, city 1 is assigned to slot 2 and city 2 is assigned to slot 1 in the solution on the left of Fig. 2. Whilst, the frequency matrix on the right side of Fig. 2 shows that city 1 is assigned to slot 1 twice and to slot 2 three times, and city 2 is assigned to slot 1 once; and so on for the other cities and slots.

In this work, entropy information theory is used to measure the diversity as follows [17]:

$$\varepsilon_i = \frac{\sum_{j=1}^{n} \frac{e_{ij}}{m} . \log \frac{e_{ij}}{m}}{-\log n}$$

$$\varepsilon = \frac{\sum_{i=1}^{n} \varepsilon_i}{n}$$

where n is the number of slots, e_{ij} is the frequency of allocating city i to slot j, and m represents the number of objects, ε_i is the entropy for city i, and ε is the entropy for one solution ($0 \le \varepsilon_i \le 1$).

We have tested the proposed HS algorithm with the multi-population mechanism (denoted as MPHSA) and without the multi-population mechanism (denoted as HSA) using the same parameter values. Table 7 lists the best and the average entropy obtained by HSA and MPHSA after 30 runs are executed on the same environmental changes where *m* = 0.5. The best results are presented in bold. The results show that applying the multi-population approach for HS

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

algorithm enhance the effectiveness in dealing with DTSP by maintaining higher diversity.

6. COMPARISON

In this section, the results obtained by the multi-population HS algorithm using multiple PARs strategy (denoted MPHSA in this section) are compared against the state-of-the-art algorithms. The compared methods are as shown in Table 8. The reference of all the three compared algorithms is 5.

The basic experimental results in DTSP with random traffic factors in case of fast environmental changes (f = 5) are given in Table 9. The best results are introduced in bold. The comparison shows that the multi-population HS algorithm is able to obtain new best results in the three tested datasets in case of small degree of dynamic changes where m = 0.1 and 0.25.

The basic experimental results for DTSP with random traffic factors in case of slow environmental changes (f = 100) are given in Table 10. The comparison shows that the proposed multi-population HS algorithm is able to obtain new best results in two out of the three tested datasets in case of small degree of dynamic changes where m = 0.1.

The basic experimental results for DTSP with cyclic traffic factors in case of fast environmental changes (f = 5) are given in Table 11. The proposed multi-population HS algorithm is able to obtain new best results in two out of the three tested datasets in case of small degree of dynamic changes where m = 0.1 and in one dataset in case of m = 0.25.

The basic experimental results for DTSP with cyclic traffic factors in case of slow environmental changes (f = 100) are given in Table 12. The results show that our proposed algorithm is able to obtain new best results in two out of the three tested datasets in case of small degree of dynamic changes where m = 0.1.

Table 13 shows statistical test results regarding the offline performance of our proposed method against RIACO, EIACO, and MIACO on random and cyclic DTSPs.

7. CONCLUSION

Hybridised HS algorithm to solve DTSP with traffic factors has been proposed in this work. The basic HS algorithm is hybridised with (i) multi-population mechanism, and (ii) multiple-PAR strategy. The hybrisisation process with the multi-population mechanism help to keep track the dynamic changes and maintain diversity and the hybrisisation with multiple-PAR strategy is suggested in order to enhance the performance on solving the DTSP.

The experiments are carried out to investigate the effect of the proposed hybrisisations on HS algorithm for the DTSP. The performance of our proposed method is verified on two variations of DTSPs with traffic factors, i.e., random and cyclic traffic delays. A number of dynamic test cases are used for both of random and cyclic types of DTSP to analyse the adaptation capabilities of the suggested algorithm. The experimental results show that the multi-population HS algorithm using multiple PARs strategy is able to outperform the basic HS algorithm and is able also to obtain a number of new best results in the DTSP with traffic factors when compared against the state-of-the-art methods.

Although the results show that the multi-population HS algorithm produces good and competitive results in case of small degree of dynamic changes, however; the results also revealed that the proposed algorithm couldn't produce better results when compared to the state-of-the-art methods in case of large degree of dynamic changes.

REFRENCES:

[1] Yang, S., Ong, Y.S. and Jin, Y. 2007. Evolutionary computation in dynamic and uncertain environments. *Ed. Springer Verlag*.

[2] Talbi, E.-G. 2009. From design to implementation. *John Wiley & Sons, Inc*.

[3] Branke, J., Salihoğlu, E. and Uyar, C. 2005. Towards an analysis of dynamic environments. In: *Proceedings of the 2005 Genetic and Evolutionary Computation Conference, ACM Press*. pp. 1433-1440.

[4] Lim, Y.-F., Hong, P.-Y., Ramli, R. and Khalid, R. 2011. An Improved Tabu Search for Solving Symmetric Traveling Salesman Problems. *IEEE Colloquium on Humanities, Science and Engineering Research (CHUSER 2011)*, Dec 5-6 2011, Penang.

[5] Mavrovouniotis, M. and Yang, S. 2013. Ant colony optimization with immigrants schemes for the dynamic travelling salesman problem with traffic factors. *Applied Soft Computing* 13, 4023-4037.

[6] Jin, Y. and Branke, J. 2005. Evolutionary optimization in uncertain environments—a survey. *IEEE Transactions on Evolutionary Computation*, 9 (3) 303-317.

[7] Cobb, H.G. 1990. An investigation into the use of hypermutation as an adaptive operator in genetic algorithms having continuous, time-

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

dependent non-stationary environments. *Technical Report AIC-90-001, Naval Research Laboratory*, Washington, USA.

[8] Nguyen, T.T., Yang, S. and Branke, J. 2012. Evolutionary dynamic optimization: A survey of the state of the art. *Swarm and Evolutionary Computation*, vol.6, pp. 1-24.

[9] Yang, S. 2008. Genetic algorithms with memory and elitism based immigrants in dynamic environments. *Evolutionary Computation* 16 (3) 385-416.

[10] Mavrovouniotis, M. and Yang, S. 2011b. Memory-based immigrants for ant colony optimization in changing environments. In: *EvoApplications 2011: Applications of Evolutionary Computing, LNCS 6624, Springer-Verlag*, pp. 324-333.

[11] Parrott, D. and Li, X. 2006. Locating and tracking multiple dynamic optima by a particle swarm model using speciation. *IEEE Transactions on Evolutionary Computation*, 10 (4): 440-458.

[12] Turky, A.M. and Abdullah, S. 2014. A multi-population harmony search algorithm with external archive for dynamic optimization problems. *Information Sciences*, Volume 272, Pages 84–95.

[13] Geem, Z.W., Kim, J.H. and Loganathan, G.V. 2001. A New Heuristic Optimization Algorithm: Harmony Search. *Simulation*, vol. 76, no. 2: 60-68.

[14] Geem, Z.W. 2010. Recent advances in harmony search algorithm. *Ed. 1 Springer Verlag*.

[15] Mansour, N., Isahakian, V. and Ghalayini, I. 2011. Scatter search technique for exam timetabling. *Applied Intelligence* 34 (2): 299-310.

[16] Geem, Z.W., Tseng, C.-L. and Park, Y. 2005. Harmony search for generalized orienteering problem: best touring in China. *Advances in Natural Computation, Lecture Notes in Computer Science*, Volume 3612, pp. 741-750.

[17] Fleurent, C. and Ferland, J. A. 1994. Genetic hybrids for the quadratic assignment problem. *Quadratic assignment and related problems*. 16: 173-187.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

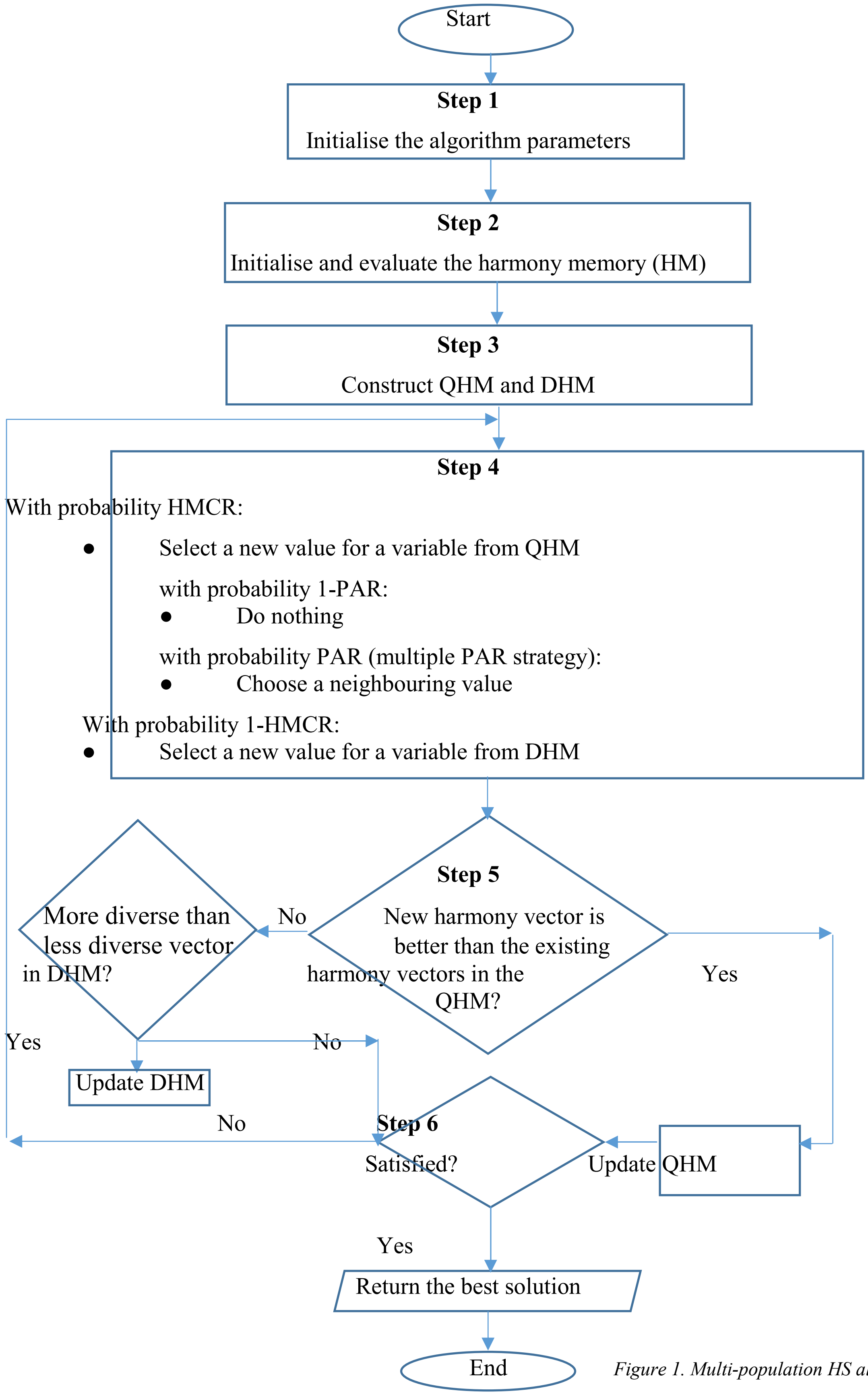

Figure 1. Multi-population HS algorithm

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 1: the experimental results of the basic HSA, MPHSA-1PAR and MPHSA-3PARs in the DTSP with random traffic factors in case of fast environmental changes (f = 5)

Algorithm	*m* = 0.1	*m* = 0.25	*m* = 0.5	*m* = 0.75
kroA100				
HSA	30372.8	40529.2	58809.6	80222.8
MPHSA-1PAR	29959.7	39959.0	55918.9	73418.5
MPHSA-3PARs	**23538.1**	**30090.3**	**42565.3**	**57246.5**
kroA150				
HSA	39049.8	45720.5	62189.4	96330.9
MPHSA-1PAR	35463.3	44068.0	62289.7	89734.5
MPHSA-3PARs	**31426.5**	**36927.4**	**58197.3**	**74281.4**
kroA200				
HSA	40014.6	50149.0	74884.2	99552.9
MPHSA-1PAR	38145.9	52336.2	69473.7	96284.4
MPHSA-3PARs	**35562.3**	**42003.9**	**65754.6**	**90203.8**
eil51				
HSA	561.5	703.8	1130.0	1425.4
MPHSA-1PAR	493.9	699.7	1105.2	1449.8
MPHSA-3PARs	**460.9**	**669.5**	**939.1**	**1308.9**
lin318				
HSA	54307.9	76439.0	106703.9	144720.4
MPHSA-1PAR	52947.2	73914.0	97658.7	135009.3
MPHSA-3PARs	**50217.5**	**69008.8**	**89797.7**	**134427.5**
pr439				
HSA	152098.0	193559.9	284835.5	357031.7
MPHSA-1PAR	138105.6	184221.7	280173.7	355169.4
MPHSA-3PARs	**130323.4**	**172824.5**	**260783.2**	**329740.5**

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

rat783				
HSA	12424.0	16220.0	22695.5	28838.9
MPHSA-1PAR	11692.1	14690.9	21906.1	28485.0
MPHSA-3PARs	**11621.1**	**14662.9**	**21604.9**	**28404.1**

Table 2: the experimental results of the basic HSA, MPHSA-1PAR and MPHSA-3PARs in the DTSP with random traffic factors in case of slow environmental changes (f = 100)

Algorithm	***m* = 0.1**	***m* = 0.25**	***m* = 0.5**	***m* = 0.75**
kroA100				
HSA	27523.9	39522.8	55579.8	75881.7
MPHSA-1PAR	28368.5	38253.1	58867.0	74265.8
MPHSA-3PARs	**23339.6**	**27249.0**	**43441.3**	**60334.4**
kroA150				
HSA	35930.0	44769.5	70032.0	91575.8
MPHSA-1PAR	35613.7	45430.9	70179.6	84943.1
MPHSA-3PARs	**29648.9**	**40503.2**	**57152.2**	**81982.1**
kroA200				
HSA	41322.2	56378.2	80714.8	101922.1
MPHSA-1PAR	38720.9	61059.7	73720.6	94213.4
MPHSA-3PARs	**34875.2**	**49570.9**	**69163.8**	**87206.1**
eil51				
HSA	593.8	774.7	1186.9	1401.8
MPHSA-1PAR	550.8	605.9	906.9	1360.7
MPHSA-3PARs	**545.1**	**560.8**	**869.3**	**1261.4**
lin318				
HSA	59830.8	76237.9	111284.1	140403.3
MPHSA-1PAR	53513.8	72771.1	105250.7	149280.1
MPHSA-3PARs	**50861.2**	**67285.8**	**101047.7**	**132607.4**

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

pr439				
HSA	146157.7	199596.8	291293.9	375393.8
MPHSA-1PAR	143572.0	192611.7	256519.2	365761.2
MPHSA-3PARs	**133953.5**	**169913.3**	**242123.7**	**348406.3**
rat783				
HSA	11650.8	16128.1	22424.2	29318.7
MPHSA-1PAR	11268.6	15944.6	21599.3	28830.1
MPHSA-3PARs	**11236.1**	**14235.7**	**20840.9**	**27918.6**

Table 3: the experimental results of the basic HSA, MPHSA-1PAR and MPHSA-3PARs in the DTSP with cyclic traffic factors in case of fast environmental changes (f = 5)

Algorithm	***m* = 0.1**	***m* = 0.25**	***m* = 0.5**	***m* = 0.75**
kroA100				
HSA	28860.3	39731.6	61814.7	70853.3
MPHSA-1PAR	27678.5	34567.1	54386.7	69962.7
MPHSA-3PARs	**22590.1**	**28655.8**	**44907.9**	**58540.6**
kroA150				
HSA	39416.4	54603.0	66016.8	90747.7
MPHSA-1PAR	35203.9	48313.7	66375.6	91324.0
MPHSA-3PARs	**29625.5**	**41759.6**	**57604.5**	**75987.6**
kroA200				
HSA	37441.1	54252.6	79317.2	103869.1
MPHSA-1PAR	37178.1	54377.1	75500.1	101488.4
MPHSA-3PARs	**33507.8**	**43268.1**	**64624.7**	**87872.2**
eil51				
HSA	706.8	775.9	995.2	1483.0
MPHSA-1PAR	508.1	759.2	982.3	1440.4
MPHSA-3PARs	**479.2**	**656.4**	**871.3**	**1270.8**

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

		lin318		
HSA	59705.8	73510.6	112984.4	140153.7
MPHSA-1PAR	56840.5	68602.4	106642.5	135648.7
MPHSA-3PARs	**52902.4**	**68357.7**	**105486.4**	**131297.6**
		pr439		
HSA	154865.1	209363.2	285646.8	377426.4
MPHSA-1PAR	139654.7	197195.6	271933.9	353211.0
MPHSA-3PARs	**134855.7**	**181477.3**	**250809.3**	**343266.0**
		rat783		
HSA	11763.1	15918.0	23155.6	29321.6
MPHSA-1PAR	11357.6	15693.2	22324.2	28801.5
MPHSA-3PARs	**11244.6**	**15168.2**	**22252.7**	**28570.6**

Table 4: the experimental results of the basic HSA, MPHSA-1PAR and MPHSA-3PARs in the DTSP with cyclic traffic factors in case of slow environmental changes (f = 100)

Algorithm	***m* = 0.1**	***m* = 0.25**	***m* = 0.5**	***m* = 0.75**
		kroA100		
HSA	33331.6	39137.7	55790.2	75586.5
MPHSA-1PAR	29372.1	37729.7	55850.0	72028.6
MPHSA-3PARs	**22095.2**	**31288.1**	**46140.6**	**62463.5**
		kroA150		
HSA	35359.0	49928.3	68859.5	97175.7
MPHSA-1PAR	35915.8	49608.8	69848.9	90789.4
MPHSA-3PARs	**28737.7**	**40186.9**	**54315.4**	**79725.4**
		kroA200		
HSA	43679.8	54835.0	83088.9	96052.6
MPHSA-1PAR	42669.8	51773.3	78504.6	95603.4
MPHSA-3PARs	**34197.6**	**47398.1**	**68647.7**	**90895.9**
		eil51		

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

HSA	544.3	781.5	1010.8	1484.4
MPHSA-1PAR	535.8	698.2	1007.3	1447.6
MPHSA-3PARs	**526.3**	**657.4**	**979.9**	**1284.3**
lin318				
HSA	66185.6	75906.3	113374.7	139509.7
MPHSA-1PAR	56205.5	71792.7	114404.9	142719.4
MPHSA-3PARs	**52237.3**	**68125.6**	**94152.9**	**132936.5**
pr439				
HSA	149845.0	198400.8	283354.8	380267.0
MPHSA-1PAR	144822.5	192192.1	269330.6	358798.3
MPHSA-3PARs	**142074.2**	**187408.5**	**267868.8**	**343560.5**
rat783				
HSA	11793.8	15716.7	23338.9	29465.6
MPHSA-1PAR	11388.2	15520.0	22479.9	28746.4
MPHSA-3PARs	**11203.3**	**15097.3**	**21074.1**	**28132.1**

Table 5: statistical test results regarding the offline performance of HSA, MPHSA-1PAR and MPHSA-3PARs on random and cyclic DTSPs tested on kroA100, kroA150 and kroA200 instances, where + or - indicates that the first algorithm or the second algorithm is significantly better.

Alg.	**kroA100**				**kroA150**				**kroA200**			
Random DTSPs												
f = 5, *m* →	0.1	0.25	0.5	0.75	0.1	0.25	0.5	0.75	0.1	0.25	0.5	0.75
MPHSA-3PARs ↔ MPHSA-1PAR	+	+	+	+	+	+	+	+	+	+	+	+
MPHSA-3PARs ↔ HSA	+	+	+	+	+	+	+	+	+	+	+	+
MPHSA-1PAR ↔ HSA	+	+	+	+	+	+	-	+	+	-	+	+
Random DTSPs												
f = 100, *m* →	0.1	0.25	0.5	0.75	0.1	0.25	0.5	0.75	0.1	0.25	0.5	0.75
MPHSA-3PARs ↔ MPHSA-1PAR	+	+	+	+	+	+	+	+	+	+	+	+
MPHSA-3PARs ↔ HSA	+	+	+	+	+	+	+	+	+	+	+	+
MPHSA-1PAR ↔ HSA	-	+	-	+	+	-	-	+	+	-	+	+

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Cyclic DTSPs												
f = 5, *m* →	0.1	0.25	0.5	0.75	0.1	0.25	0.5	0.75	0.1	0.25	0.5	0.75
MPHSA-3PARs ↔ MPHSA-1PAR	+	+	+	+	+	+	+	+	+	+	+	+
MPHSA-3PARs ↔ HSA	+	+	+	+	+	+	+	+	+	+	+	+
MPHSA-1PAR ↔ HSA	+	+	+	+	+	+	-	-	+	-	+	+

Cyclic DTSPs												
f = 100, *m* →	0.1	0.25	0.5	0.75	0.1	0.25	0.5	0.75	0.1	0.25	0.5	0.75
MPHSA-3PARs ↔ MPHSA-1PAR	+	+	+	+	+	+	+	+	+	+	+	+
MPHSA-3PARs ↔ HSA	+	+	+	+	+	+	+	+	+	+	+	+
MPHSA-1PAR ↔ HSA	+	+	-	+	-	+	-	+	+	+	+	+

Table 6: statistical test results regarding the offline performance of HSA, MPHSA-1PAR and MPHSA-3PARs on random and cyclic DTSPs tested on eil51, lin318, pr439 and rat783 instances, where + or - indicates that the first algorithm or the second algorithm is significantly better.

Alg.	eil51				lin318				pr439				rat783			
Random DTSPs																
f = 5, *m* →	0.1	0.25	0.5	0.75	0.1	0.25	0.5	0.75	0.1	0.25	0.5	0.75	0.1	0.25	0.5	0.75
MPHSA-3PARs ↔ MPHSA-1PAR	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
MPHSA-3PARs ↔ HSA	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
MPHSA-1PAR ↔ HSA	+	+	+	-	+	+	+	+	+	+	+	+	+	+	+	+

Journal of Theoretical and Applied Information Technology
31st January 2017. Vol.95. No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Random DTSPs

f = 100, *m* →	0.1	0.25	0.5	0.75	0.1	0.25	0.5	0.75	0.1	0.25	0.5	0.75	0.1	0.25	0.5	0.75
MPHSA-3PARs ↔ MPHSA-1PAR	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
MPHSA-3PARs ↔ HSA	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
MPHSA-1PAR ↔ HSA	+	+	+	+	+	+	+	-	+	+	+	+	+	+	+	+

Cyclic DTSPs

f = 5, *m* →	0.1	0.25	0.5	0.75	0.1	0.25	0.5	0.75	0.1	0.25	0.5	0.75	0.1	0.25	0.5	0.75
MPHSA-3PARs ↔ MPHSA-1PAR	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
MPHSA-3PARs ↔ HSA	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
MPHSA-1PAR ↔ HSA	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+

Journal of Theoretical and Applied Information Technology
31st January 2017. Vol.95. No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

	Cyclic DTSPs															
f = 100, *m* →	0.1	0.25	0.5	0.75	0.1	0.25	0.5	0.75	0.1	0.25	0.5	0.75	0.1	0.25	0.5	0.75
MPHSA-3PARs ↔ MPHSA-1PAR	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
MPHSA-3PARs ↔ HSA	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
MPHSA-1PAR ↔ HSA	+	+	+	+	+	+	-	-	+	+	+	+	+	+	+	+

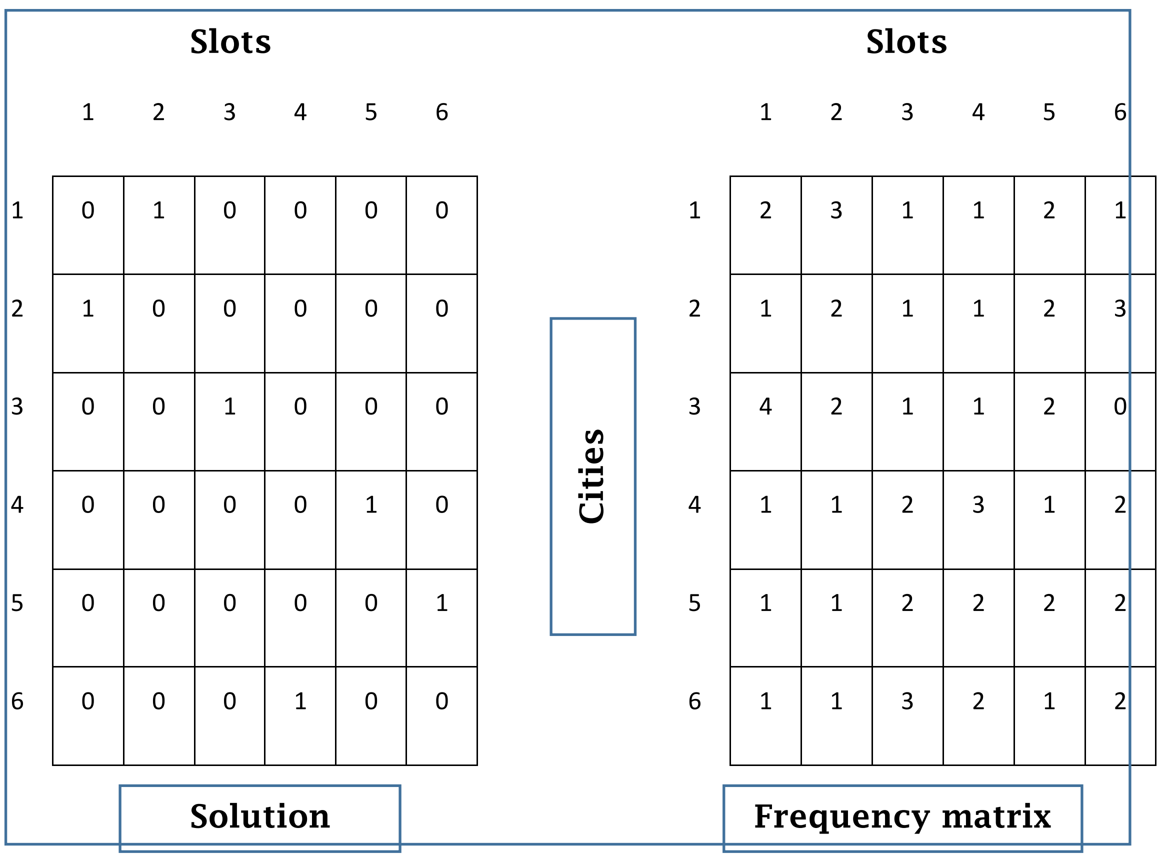

Figure 2. solution and its identical frequency matrix.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 7: the best and average entropy results of the basic HSA and MPHSA in the random DTSP with traffic factors

Algorithm	*f* = 5 Best	*f* = 5 Average	*f* = 100 Best	*f* = 100 Average
kroA100				
HSA	0.3889	0.3881	0.3890	0.3882
MPHSA	**0.3944**	**0.3911**	**0.3938**	**0.3910**
kroA150				
HSA	0.2453	0.2447	0.2451	0.2443
MPHSA	**0.2473**	**0.2462**	**0.2479**	**0.2457**
kroA200				
HSA	0.1761	0.1757	0.1763	0.1760
MPHSA	**0.1781**	**0.1770**	**0.1775**	**0.1770**

Table 8: the state-of-the-art methods

#	Description	Symbol
1	Random immigrants ant colony optimisation	RIACO
2	Elitism-based immigrants ant colony optimisation	EIACO
3	Memory-based immigrants ant colony optimisation	MIACO

Table 9: the experimental results in the DTSP with random traffic factors in case of fast environmental changes (f = 5)

Algorithm	*m* = 0.1	*m* = 0.25	*m* = 0.5	*m* = 0.75
kroA100				
MPHSA	**23538.1**	**30090.3**	42565.3	57246.5
RIACO	26557.8	30420.4	38252.7	**53471.6**
EIACO	26100.0	30258.0	**38166.9**	53491.6
MIACO	26198.9	30341.5	38312.4	53728.9
kroA150				
MPHSA	**31426.5**	**36927.4**	58197.3	74281.4
RIACO	33816.5	38512.4	48133.7	**66340.4**
EIACO	33376.1	38272.3	**48053.8**	66428.3
MIACO	33511.3	38382.4	48243.0	66674.8

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

		kroA200		
MPHSA	**35562.3**	**42003.9**	65754.6	90203.8
RIACO	38535.5	43914.6	55048.9	**76008.3**
EIACO	37926.2	43576.3	**54922.2**	76103.4
MIACO	38050.7	43677.9	55108.7	76355.5

Table 10: the experimental results for the DTSP with random traffic factors in case of slow environmental changes (f = 100)

Algorithm	***m* = 0.1**	***m* = 0.25**	***m* = 0.5**	***m* = 0.75**
		kroA100		
MPHSA	**23339.6**	27249.0	43441.3	60334.4
RIACO	23635.8	25846.7	32876.3	44905.7
EIACO	23417.2	**25660.5**	**32576.1**	**44339.0**
MIACO	23398.7	25736.8	32687.5	44458.1
		kroA150		
MPHSA	**29648.9**	40503.2	57152.2	81982.1
RIACO	30343.8	33976.4	40704.9	56085.0
EIACO	29892.6	**33280.7**	**39994.4**	**54963.4**
MIACO	29893.0	33443.3	40175.3	55175.1
		kroA200		
MPHSA	34875.2	49570.9	69163.8	87206.1
RIACO	34203.1	37766.3	46036.8	65169.9
EIACO	**33496.6**	**37072.2**	**45106.1**	**63347.5**
MIACO	33576.9	37236.1	45206.7	63781.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 11: the experimental results for the DTSP with cyclic traffic factors in case of fast environmental changes (f = 5)

Algorithm	***m* = 0.1**	***m* = 0.25**	***m* = 0.5**	***m* = 0.75**
		kroA100		
MPHSA	**22590.1**	**28655.8**	44907.9	58540.6
RIACO	25757.7	29990.3	37189.3	45013.2
EIACO	23991.4	29530.6	37213.1	45013.2
MIACO	24106.7	29586.5	**37001.2**	**44630.3**
		kroA150		
MPHSA	**29625.5**	41759.6	57604.5	75987.6
RIACO	33389.7	37091.3	42240.5	56595.2
EIACO	30671.2	**36228.3**	**41787.5**	56720.6
MIACO	30767.2	36337.4	41829.0	**56243.3**
		kroA200		
MPHSA	33507.8	43268.1	64624.7	87872.2
RIACO	36192.8	41550.4	49383.7	61938.5
EIACO	33169.1	**40329.2**	**48421.4**	61754.9
MIACO	**33164.1**	40375.5	48565.4	**61524.3**

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 12: the experimental results for the DTSP with cyclic traffic factors in case of slow environmental changes (f = 100)

Algorithm	*m* = 0.1	*m* = 0.25	*m* = 0.5	*m* = 0.75
kroA100				
MPHSA	**22095.2**	31288.1	46140.6	62463.5
RIACO	23980.4	26401.9	31072.9	37717.3
EIACO	23220.3	26061.0	30988.1	37486.1
MIACO	23272.8	**26031.9**	**30850.6**	**37361.5**
kroA150				
MPHSA	**28737.7**	40186.9	54315.4	79725.4
RIACO	30245.7	32323.3	36056.6	47116.3
EIACO	29558.5	**32012.2**	35761.9	46395.4
MIACO	29732.8	32035.5	**35745.8**	**46251.5**
kroA200				
MPHSA	34197.6	47398.1	68647.7	90895.9
RIACO	33641.2	35914.9	42646.5	52493.2
EIACO	**32381.9**	35516.8	**41779.5**	51617.5
MIACO	32558.8	**35450.6**	41842.3	**51594.0**

Journal of Theoretical and Applied Information Technology
31st January 2017. Vol.95. No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 13: statistical test results regarding the offline performance of MPHSA against the state-of-the-art methods on random and cyclic DTSPs tested on kroA100, kroA150 and kroA200 instances, where + or - indicates that the first algorithm or the second algorithm is significantly better.

Alg.	kroA100				kroA150				kroA200			
Random DTSPs												
f = 5, m →	0.1	0.25	0.5	0.75	0.1	0.25	0.5	0.75	0.1	0.25	0.5	0.75
MPHSA-3PARs ↔ RIACO	+	+	-	-	+	+	-	-	+	+	-	-
MPHSA-3PARs ↔ EIACO	+	+	-	-	+	+	-	-	+	+	-	-
MPHSA-1PAR ↔ MIACO	+	+	-	-	+	+	-	-	+	+	-	-
Random DTSPs												
f = 100, m →	0.1	0.25	0.5	0.75	0.1	0.25	0.5	0.75	0.1	0.25	0.5	0.75
MPHSA-3PARs ↔ RIACO	+	-	-	-	+	-	-	-	-	-	-	-
MPHSA-3PARs ↔ EIACO	+	-	-	-	+	-	-	-	-	-	-	-
MPHSA-1PAR ↔ MIACO	+	-	-	-	+	-	-	-	-	-	-	-
Cyclic DTSPs												
f = 5, m →	0.1	0.25	0.5	0.75	0.1	0.25	0.5	0.75	0.1	0.25	0.5	0.75
MPHSA-3PARs ↔ RIACO	+	-	-	-	+	-	-	-	+	-	-	-
MPHSA-3PARs ↔ EIACO	+	-	-	-	+	-	-	-	-	-	-	-
MPHSA-1PAR ↔ MIACO	+	-	-	-	+	-	-	-	-	-	-	-
Cyclic DTSPs												
f = 100, m →	0.1	0.25	0.5	0.75	0.1	0.25	0.5	0.75	0.1	0.25	0.5	0.75
MPHSA-3PARs ↔ RIACO	+	-	-	-	+	-	-	-	-	-	-	-
MPHSA-3PARs ↔ EIACO	+	-	-	-	+	-	-	-	-	-	-	-
MPHSA-1PAR ↔ MIACO	+	-	-	-	+	-	-	-	-	-	-	-