2018 Tenth International Conference on Advanced Computational Intelligence (ICACI)
March 29–31, 2018, Xiamen, China

A Random Distribution Harmony Search Algorithm

Jiadong Gu and Defeng Wu*
School of Marine Engineering, Jimei University, Xiamen, 361021, China
15757164910@163.com, arcwdf@gmail.com

***Abstract*—Harmony Search (HS) algorithm is widely applied in high-dimensional complex optimization problems. However, there are several drawbacks with HS algorithm for which HS algorithm is unable to find global optimumsolution efficiently and effectively sometimes. In this paper, a random distribution Harmony Search (RDHS) algorithm which subjoins a random distribution operation to HS was proposed. The random distribution operation can increase convergence speed effectively as well as optimum ability. The effectiveness and advantages are demonstrated through several simulationexperiments of conventional test functionscompared with HS and improved Harmony Search (IHS).**

Keywords—Harmony Search Algorithm; Improved Harmony Search algorithm; Random Distribution Harmony Search; Test function

I. Introduction

With the expansion of human living environmentas well as the transformation and understanding toward the scope of nature broaden, realistic problems have been more sophisticated to deal with.The conventional optimization methodssuch as Newton method, simplex method, optimal gradient method, pattern search method is unable to meet the needs of complex optimization problems. Therefore,efficient optimization algorithms have become one of the research goals for scientific researchers. At the same time, the wide application of high-speed computer also provides a useful tool to solve complex problems [1].

Thereis a large amount of complex and high-dimensional optimization problems in every field of our national economy, such as optimal dispatch in transportation, optimal allocation of resources, optimal layout of production process, optimal development of land, rational distribution of crops. intelligent optimization algorithm are effective methods to solve them In many instances [2], [3].

The researchof intelligent optimization algorithm is a very activefieldinrecentyears.Compared with conventional optimization algorithm,themultiply dimensional complex optimization problem can be solved more effectively relied on high-speed computers based on intelligent optimization algorithm. Some excellent intelligent algorithms can effectively avoid the local optimal trap that the conventional optimization method would easily trap into. There are many intelligent optimization algorithms such as Genetic Algorithm, Simulated Annealing algorithm,Tabu Searchalgorithm, Artificial Bee Colony [4],Particle Swarm Optimization, Harmony Search (HS) algorithm and so on.

HS is a new meta-heuristic algorithm which is inspired by music improvisation process [5].In the musical performance, the music masterwould record some wonderful harmonic combinations in their playing process by their own memories and then formed their own harmony memories. And then in the subsequent performance, constantly updated their harmony memories. Finally, they would achieve a set of wonderful harmony. The algorithm can be applied in engineering as well. Through high speed computers,The better solution in the calculation process is recorded to form their own harmony memories and then various ways are applied to increase the diversity of the solution, so that the algorithm is not easy to trap into the local optimal solution. Finally, worse solutionin memories is eliminated gradually to make the whole library reach a better level [5], [9].

In recent years, the research and improvement of Harmony Search algorithm is also in full swing all around the world. Some related works on the HS improvement were proposed. For instance, Harmony Search algorithm through changing Harmony Search algorithm PAR parameter and bandwidth based on the dynamic number of iterations was proposed in[6], namelyImprove Harmony Search(IHS) ,and then it was applied to the schedule of multiple generators. A series of improvement measures such as Improved Harmony Search, Self-adaptive HarmonySearch,Global-best Harmony Search, Self-adaptiveGlobal-bestHarmony Search, NovelGlobal HarmonySearch are proposed for the Harmony Search algorithm,and the performance of different improved methods is tested in [7]. An adaptive trimming factor was added to Harmony Search and then this method was applied to solve geometricconstraint problems in [8]. Selection, mutation and crossover operations in Genetic Algorithm were added to Harmony Search algorithm,which improved the diversity of harmony memories and this method was appliedtovehicle path planning in [9]. Two famous algorithms, Particle Swarm Optimization (PSO) and Harmony Search algorithm (HS) were combined to form the algorithm called Hybrid Multi-Swarm With Harmony Search Algorithm(HMSHS) to get better algorithm performance in[10] and a large amount of experiments have been made to compare performance with Dynamic Multi-Swarm Particle Swarm Optimizer Harmony Search(DMS-PSO-HS) in [11].

The remainders of the paper are organized as follows. The original Harmony Search and two enhance versions of HS are involved in SectionII. Several conventional test functions are introduced in Section III, Meanwhile, single performance tests of HS, IHS and RDHS on various test functions and a large number of experiments to evaluate performance of HS, IHS, RDHS are displayed in this Section as well. Finally, Conclusion of this paper is proposed in Section 4.

II. PROPOSED ALGORITHM

A. Harmony Search Algorithm

The original Harmony can be roughly divided into 5 steps after being summarized:

Step 1: Determined the objective function, the constraint ranges and the dimensions of each solution (D). Set Harmony Memory Considering Rate (*HMCR*), Pitch AdjustingRate (*PAR*), Harmony Memory Size (*HMS*), Disturbance bandwidth (*bw*), and Iteration times (*Iter*).

Step 2:Initialize harmony memory. Initial solution was put into harmonymemories.

Define HM as harmony memory, it can be expressed as follow:

$$\text{HM}=\left|\begin{array}{cccc|c} x_1^1 & x_2^1 & \dots & x_D^1 & f(X^1) \\ x_1^2 & x_2^2 & \dots & x_D^2 & f(X^2) \\ \dots & \dots & \dots & \dots & \dots \\ x_1^{HMS} & x_2^{HMS} & \dots & x_D^{HMS} & f(X^{HMS}) \end{array}\right| \tag{1}$$

Where X^j is *No.j* solution, x_i^j is *No.i* component of *No.j* solution and $f(X^j)$ is the function value of *No.j* solution .

Step 3: Standby solution $x' = \{x'_1, x'_2 \dots x'_D\}$ was generated by following three ways:

- Random selection in harmony memory.
- Initialize solution directly.
- Generate new solution through perturbation trimming.

Fine tuning is operated by the following formula.

$$x_i^j = x_i^j \pm \mu * p \tag{2}$$

Step 4: Compare standby solution x' with the worst solution in HM, if the former is superior to the latter, and then updates *HM* with this standby solution.

Step 5: Repeat step 3,4. And the optimization process is stopped until the criterion is satisfied.

B. Improve Harmony Search (IHS) Algorithm[4]

The performance of HS was improved by dynamically changing two parameters in HS based on the number of iterations. These two parameters are Pitch Adjusting Rate (*PAR*) and Disturbance bandwidth (*bw*) and *N* in following equations represents iteration Number. The improved method is shown by equations (3), (4) and (5):

$$PAR(ite) = PAR(min) + (PAR(max) - PAR(min)) * Iter / N \tag{3}$$

$$\mu(ite) = \mu(max) * e^{c*Iter} \tag{4}$$

$$c = \ln(\frac{\mu(\min)}{\mu(\max)})/\mathrm{N} \tag{5}$$

C. Random Distribution Harmony Search Algorithm

A Random Distribution Harmony Search algorithm is proposed in this paper which adds a parameter Assign as the probability of random distribution. A random distribution operation is added to step 3 of HS. The main process is shown by Fig1 as follow:

```
Initialize Harmony Search ( HM );
Generate New Solution x^new = {x_1^j, x_2^j ... x_D^j};
Random Distribution Operation On Solution x^new = {x_1^j, x_2^j ... x_D^j};
for( i : length( x^j ))
a = randi([ 1, length( x^j )])
r = rand()
t = x_i^j
x_i^j = x_i^j + x_a^j * r
if( x_i^j >= C_i )
x_i^j = C_i
x_a^j = x_a^j - ( C_i - t )
endif
else
x_a^j = x_a^j * ( 1 - r )
endelse
endfor
Compare new solution with worst solution ;
Choose a better Solution to use;
```

Fig.1.Pseudocode of Random Distribution Operation

In Fig.1, C represents constraintssequencewhich can unable tobecrossed. *length*(x^j) represents the capacity of x^j (No.*j* solutionin HM), r is assigned to random number from 0-1 and a is an integer from 1 to *length*(x^j).

This operation make significant effect on the diversity of HM, making the diversity of the algorithm not only rely on the initial random,In principle, Convergence speed and global search capabilities of algorithm will be enhanced with this operation. In the later parts of this paper, this conclusion will be tested and specifically analyzed.

III. TEST FUNCTIONS AND EXPERIMENTS

All test functions are shown in TABLE I, including function range, number of arguments, extreme value and the exact form of the function where Range represents the initialization ranges, D represents the dimensions of each solution as follows:

$$\mathrm{D} = length(x^j) \tag{6}$$

Function is the name of function, *Min* is the global optimum solution of function.

$$Min = minf(x^1, x^2, \dots x^j) \tag{7}$$

TABLE I. GENERAL INTRODUCTION TO TEST FUNCTIONS [7]

No	Test Function Introduction			
	Range	***D***	***Min***	***Function***
1	[-5.25,5.25]	5	25	Stepint
2	[-10,10]	10	0	Sphere
3	[-5.25,5.25]	8	0	Rastrigin
4	[-4.5,4.5]	2	0	Beale
5	[-600,600]	8	0	GireWank
6	[-5,5]	2	-1.032	6 hump Camelback
7	[-32,32]	30	0	Ackley
8	[-100,100]	2	0	Schaffer
9	[-10,10]	8	0	Dixon-Price
10	[0,pi]	2	-1.801	Michalewicz2
11	[-1.28,1.28]	30	0	Quartic
12	[-10,10]	30	0	Schewel2

A. Single test

Two test functions are selected in this section for experiment: 8-demonsional Rastrigin and 30-Demonsional Quartic. And the parameters are shown in TABLE II as follows:

TABLE II. PARAMETERS FOR EXPERIMENT[6]

Parameters Setting			
Harmony Memory Considering Rate(HMCR)	***Pitch Adjusting Rate(PAR)***	***Harmony Memory Size(HMS)***	***Disturbance bandwidth(bw)***
0.9	0.7	10	0.1

1) Experiment About HS and RDHS

Two single tests about HS and RDHS on different functions are shown in these figures. Firstly, we choose 10000 as number of iterations for optimization on Quartic function is easier than Rastrigin function. As for Rastrigin function, we select 20000.

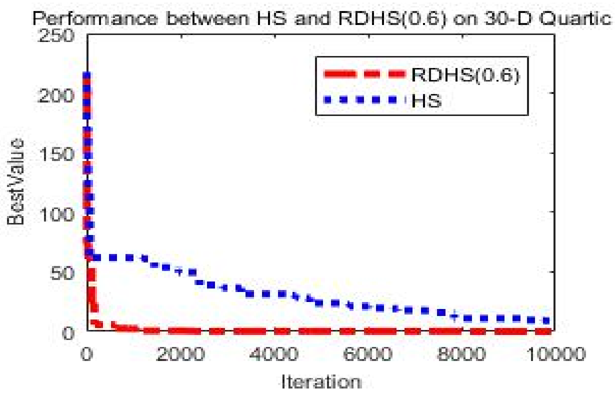

Fig.2.performance between HS and RDHS(0.6) on 30-D Quartic

Through Fig.2and Fig.3, After adding random distribution into HS, Convergence speed and optimum ability were

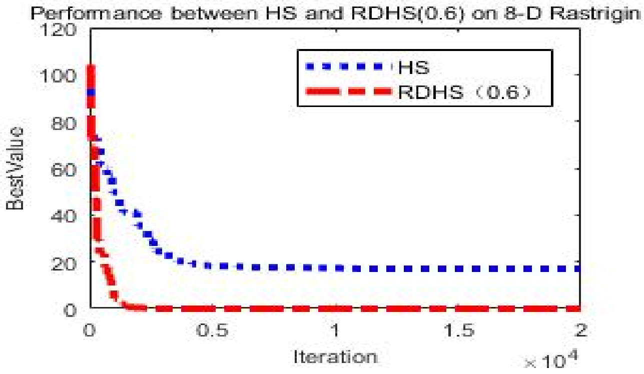

Fig.3.performance between HS and RDHS(0.6) on 8-D Rastrigin

enhanced significantly. About 100 iterations on Quartic, RDHS has got the best solution's field and then better solution could be found by fine-tuning. However, convergence speed of HS is slower, after 10000 iterations, HS starts to get the best solution's field. This is to say, convergence speed and optimum ability of RDHS is better than HS. We can also learn that HS can't get best solution field sometimes from Fig.2 while RDHS can. As for the ability to jump out of the local optimum solution, RDHS performed better as well.

2) Experiment About HS and IHS

IHS is an improve version for HS which has dynamic parameters based on iteration number. The parameters used in this paper are shown in TABLE III as follows:

TABLE III. PARAMETERS FOR IMPROVED HARMONY SEARCH [6]

Parameters Setting			
Maximum Pitch Adjusting Rate(PAR_{max})	***Minimum Pitch Adjusting Rate(PAR_{min})***	***Maximum Disturbance bandwidth(bw_{max})***	***Minimum Disturbance bandwidth(bw_{min})***
0.9	0.1	0.1	0.01

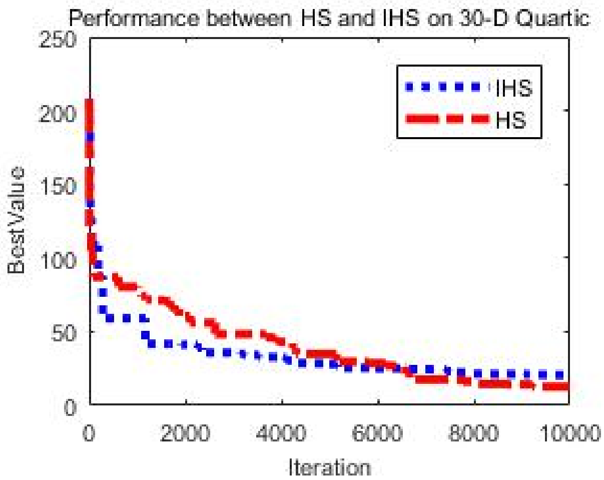

Fig .4.performance between HS and IHS on 30-D Quartic

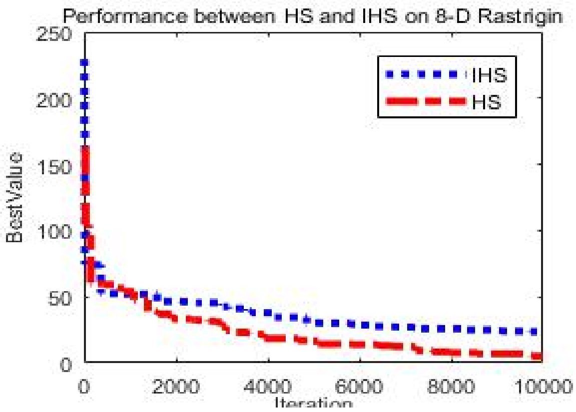

Fig .5.performance between HS and IHS on 8-D Rastrigin

From Fig. 4 and Fig.5, we can find IHS is instable for the promotion of HS and is limited to the certain function.

3)Performance about RDHS while Assign is from 0 to 1

As is shown in Fig.6, performance of algorithm is different with different *Assigns*, mainly reflects on convergence speed. However, RDHS always performs better than HS no matter what value of *Assign* is. However, performance of algorithm is not proportional to the value of *Assign*. From this figure, 0.6 can be found is the best value of Assign to improve performance of HS, both on convergence speed and best solution's quality.

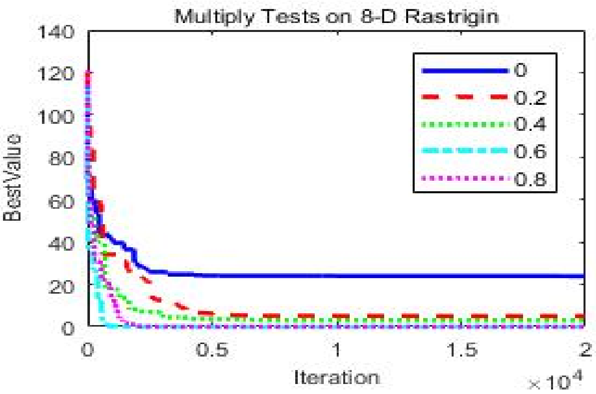

Fig .6.Multiply Tests on 8-D Rastrigin

B. *Multiple Tests AndStatistics*

1) Introduction of algorithm stability

An important indicator evaluating pros and cons of an algorithm lies in the probability that the algorithm can obtain the optimal solution after several operations,namely,the stability of algorithm.It can be seen from the previous section that RDHS has some advantages over HS in single-function test. In this section, mean value,standard deviationand optimalsolution of each algorithm were multiple calculated on different test functions. The merits of the algorithm are more convincing than single-function test.After 20 calculations, a series date can be obtained to show different algorithms' performance.

2) Parameters settings and statistics display

- *Min*: Global minimum.
- *Mean*: Mean value of 20 repeated times
- *StdDev*: Standard deviation of 20 repeated times.
- *MinValue*: Minimum of 20 repeated times.

TABLE IV. MULTIPLE CALCULATIONS DISPLAY

Multiple calculations display				
Function	***Result***	***HS***	***IHS***	***RDHS(0.6)***
Stepint	Mean	25.2491	25.5619	25.0104
	StdDev	0.2216	0.3325	0.0089
	MinValue	25.0344	25.0301	25.0005
Sphere	Mean	1.21	1.7778	1.51E-05
	StdDev	1.21	1.7333	0.6198
	MinValue	0.002	0.0138	5.65E-08
Rastrigin	Mean	21.9157	23.542	2.1394
	StdDev	8.3417	6.3868	2.1228
	MinValue	8.0039	10.884	4.85E-04
Beale	Mean	7.19E-04	5.54E-04	6.91E-04
	StdDev	0.0012	3.77E-04	0.0012
	MinValue	9.90E-06	2.11E-05	2.07E-06
GireWank	Mean	20.3182	19.5325	0.8631
	StdDev	6.819	5.5678	1.5276
	MinValue	8.2273	8.0325	0.0099
6 Hump Camelback	Mean	-1.03	-1.03	-1.0316
	StdDev	3.05E-06	2.72E-07	3.28E-07
	MinValue	-1.03	-1.03	-1.0316
Ackley	Mean	18.7718	18.8842	2.5106
	StdDev	0.2863	0.1906	2.5618
	MinValue	18.1914	18.6108	0.0098
Schaffer	Mean	0.012	0.0097	0.0068
	StdDev	0.0089	1.63E-09	0.0046
	MinValue	2.29E-06	0.0097	9.79E-10
Dixon-Price	Mean	130.201	162.9045	1.8991
	StdDev	126.2081	6.5931	4.193
	MinValue	2.4109	144.4104	0.0091
Michalewicz2	Mean	-1.8031	-1.8031	-1.8031
	StdDev	2.82E-05	2.76E-06	5.56E-05
	MinValue	-1.8013	-1.8013	-1.8013
Quartic	Mean	0.5329	5.2762	0.0257
	StdDev	0.0943	1.7584	0.01
	MinValue	0.3488	2.3188	0.0098
Schewel2	Mean	1005.8	824.1317	0.6201
	StdDev	1.664E-13	2.3328E-	0.2816

Multiple calculations display				
Function	***Result***	***HS***	***IHS***	***RDHS(0.6)***
	MinVlaue	1005.8	824.1317	0.1405

As TableIV shows,after20calculations on 12 test functions, we have got their mean, StdDev and MinValue by whichwe can compare different performance of HS, IHS and RDHS. Obviously, after adding random distribution into HS, appealthree indicators get better. Typical data is*8-D* Rastrigin, *10-D* Dixon-Price.

3) *scatter plots of Multiple tests*

Six scatter plots about performance of RDHS on several test functions 20 times while the value of Assign is set as 0, 0.3, 0.6 and 0.9 are displayed in this section. These six conventional test functions are Quartic, Stepint, Rastrigin, GireWank, Dixon and Ackley in TABLE IV.

From these scatter plots, Obviously, the probability for algorithm to find the optimal solution can been increased by random distribution. Moreover, Harmony Search algorithm

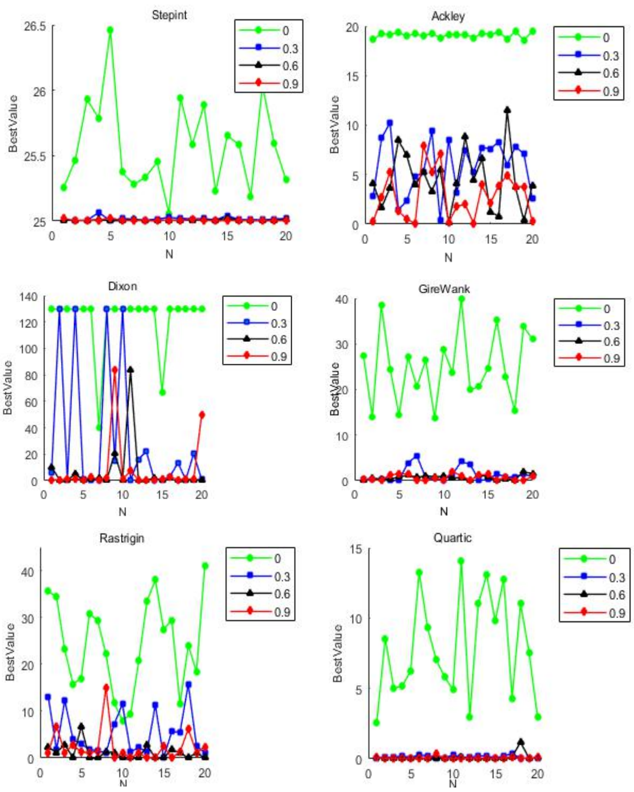

Fig .7.Scatter plots of Multiple tests

often trapped into the local optimal solution of the function and could not jump out of the local optimal solution with few iterations for lackingdiversity in HM for Diversity of HM in HS is barely expanded by initiation. Therefore, the ability to jump out of the local optimal solution of HS will be extremely limited while the number of iterations is low. Conventional figures are calculations about Dixon-Price and Ackley. After 20 calculations, it's obvious that HS has been trappedinto local optimal solution. So, Although the solution is stable, they are not global optimal solution. Compared to HS, RDHS enhances the whole diversity of HM with random distribution for which the ability of RDHS to jump out of local optimal solution become excellent. However, algorithm's stability is influenced by this operation sometimes. Convention figures are calculations about Dixon-Price and Rastrigin. Although the search ability of the global optimal solution is increased, the solution has a great fluctuation and is not stable enough. Therefore, follow-up research will meet the need to overcome this shortage which is also our main problem for future works.

IV. CONCLUSION

HS was enhanced in this paper by adding random distribution, after several tests on various functions once or 20 times, two conclusions can be summarized: The global search ability of HS is not excellent when the number of iterations is low. After adding the operation called random distribution to HS, diversity of harmony memory can be effectively improved. Thus, the convergence speed and ability of global searching can be improved in a way. Appropriate value of *Assign* is essential to RDHS, because an oversized *Assign* value may destroy original state of harmony memory which is disadvantageous to the algorithm optimization. For instance, Rastrigin and Quartic functions have the best results while *Assign* is 0.6 instead of 0.8 or 0.9.

V. ACKNOWLEDGMENT

This work was financially supported by National Natural Science Foundation of China (51249006)

REFERENCES

[1] W. Xing and J.X. Xie, Modern Optimization Calculation Method, Beijing:Tsinghua University Press,2001.

[2] L. Wang,Intelligent Optimization Design And Applications, Beijing:Tsinghua University Press, 2001.

[3] N. Pathak, M. Mishra, S. Pratap, and S, Kushwah,"Improved Local Search Based Modified ABC Algorithm for TSP Problem" 2017 4th International ConferenceOnElectronic And Communication Systems(ICECS), pp. 173-178, 2017

[4] D. Karaboga and B. Akay, "A comparative study of Artificial Bee Colony algorithm", Applied Mathematics and Computation, vol. 214, pp. 108-132, 2009.

[5] K. S. Lee and Z. W. Geem ,"A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice," Computer Methods in Applied Mechanics and Engineering, vol. 174, pp. 3902-3933, 2005.

[6] P. Yadav, R. Kumar, S. K. Panda , and C. S. Chang, "An improved Harmony Search algorithm for optimal scheduling of the diesel generators in oil rig platforms," Energy Conversion And Management,vol. 52, pp. 893-903, 2011.

[7] M. Castelli, S. Silva,L. Monzoni, and L. Vanneschi,"Geometric Selective Harmony Search," Information sciences, vol. 279, pp. 468-482, 2014.

[8] L.J. Sun, W.M. Quan, and Q. Zhao, "Research on adaptive fine-tuning disturbance harmony search algorithm forgeometric constraints solving," Application Research of computer, vol.27, pp. 2773-2779, 2010

[9] Y. Bo, W. L. Wang, and Y. L. H. Wang, "Improve Genetic Harmony Algorithm and Application in Vehicle Routing, "Computer MeasurementAndControl, vol. 192, pp. 3068-3071, 2011.

[10] W. Phuchan, B. Kruatrachue, and K. Siriboon,"Hybird Multi-Swarm with Harmony Search Algorithm," 2017 14th International Conference OnElectrical Engineering/Electronic,Computer Telecommunications And Information Technology, pp. 541-544, 2017.

[11] S. Z. Zhao, P. Nagaratnam S. Snr, and Swagatam Das, “Dynamic Multi-Swarm Particle Swarm Optimizer with Sub-regional Harmony Search,” pp. 1-8, 2010.