# The Selection of Particle Swarm Optimization Learning Factors Values in Solving the Multiple Travelling Salesman Problem

**Belal Al-Khateeb[1]**

1 College of Computer Science and Information Technology, University of Anbar, Ramadi, Iraq, Corresponding Author Email: belal@computer-college.org.

**Abstract :**This paper addresses the question of whether fixed learning (acceleration) factors are an important factor in the Particle Swarm Optimization (PSO) by testing many selected values for those factors and apply them in solving the Multiple Traveling Salesman Problem (MTSP). Extensive experiments are done and those experiments show that the learning factors are problem dependent, therefore it is recommended to do the same experiments that are done in this paper for each problem that intend to be solved by PSO.

## I. Introduction

Particle Swarm Optimization (PSO) is one of the most popular and successful nature-inspired optimization algorithms. PSO is described as a flock of birds flying randomly to find a place of food. The best one becomes the leader of the flock and all other birds must follow it. Each bird represents a solution in a search space called ‘particle’. PSO is initialized as random particles, then updating generations to searching of optima [1].

PSO helps to find optimal solutions of optimization problems that have continuous distinctions between variables [2]. With the passage of time, it was developed to deal with discrete problems. The use of PSO algorithm has been successful in most problems that deal with discrete and continuous problems. Therefore, this algorithm has been developed by researchers to be able to solve problems that contain more than one objective (multi-objective)[3].

Many previous researches used metaheuristics and applied artificial techniques in solving real life problems, those researches used parameters selection techniques for the applied algorithms [4][5][6][7][8].

In this research, a set of extensive experiments are done for the first time to find the best learning (acceleration) factors for PSO in order to solve the MTSP efficiently. The designed experiments show that the learning factors are problem dependent therefore; it cannot be fixed for all problems.

The rest of the paper is organized as follows; in section II, related work is presented. PSO is discussed in section III. Section IV presents MTSP. Section V shows the experimental setup for this work. Section VI presents the obtained results and the conclusions are presented in Section VII.

## II. Literatures Review

Several researches are done for PSO parameters selection, those researches mostly focused on the choosing the best values for the inertia weight. Feng et.al. proposed an adaptive, easy to implement and low cost inertia weight strategy [9], this strategy depends on particle’s position and velocity rather than the number of process iterations. This was done by the illumination of Butterworth filter. The obtained results show that, with careful parameters settings, the proposed strategy was successful and it can be used for many applications.

Bansal et.al. [10] studied 15 popular Inertia Weight strategies and compares their performance on five optimization test problems in order to show the importance of the inertia weight for the PSO exploration and exploitation. The obtained results show that Chaotic Inertia Weight is the best strategy for better accuracy. Random Inertia Weight strategy is best for better efficiency.

Chauhan et.al. [11] proposed a three novel inertia weight strategies, the first strategy is based on the adaptive decreasing of the inertia weight with the iteration number, while the second and third strategy are based on Gompertz function. The obtained results showed that those strategies enhanced the performance quality and convergence rate of PSO.

ISSN 1943-023X

**Received: 15 May 2018/Accepted: 20 June 2018**

Maca and Pech [12] presented updating random strategies for inertia weight; those strategies are based on beta distribution. The obtained results show that the presented strategies can enhance the PSO exploration and exploitation.

Harrison et.al. [13] applied 18 inertia weight control strategies and found that only the random selection of the inertia weight can outperform the fixed value inertia weight. For more in depth review for the inertia weight control strategy, readers can refer to the work of Rathore and Sharma [14].

The majority of previous researches didn't consider the selection of the learning (acceleration) factors in PSO, so this paper aims to be the first research to address this issue.

## III. Particle Swarm Optimization

The Particle Swarm Optimization (PSO) algorithm is one of computational algorithms that are inspired from animals' behavior such as bird flocks [15] and fish schools [16]. PSO is a population based search algorithm, simple in implementation, effective and considered as a global optimization algorithm [1]. It requires only initialization of mathematical operators. In addition, it is inexpensive in both speed and memory requirements. PSO was developed by Kennedy and Eberhart in 1995. Compared to other evolutionary algorithms, PSO was found to have a unique concept which was a particle (potential solutions) flying in search space, accelerating toward better solutions and has ability to find a feasible solution quickly [17].

The swarm of PSO consists of particles. Each particle represents a potential solution in optimization problem. The particles have two main attributes that are position and velocity. The position of each particle is updated according to its own experience and the experience of its neighbors. The velocity is adjusted to determine the direction that a particle needs to move.

During swarm movement, a particle updates its position depending on new velocity and previous position that obtained by the experiments in search space, while the updating of particle's velocity depends on previous velocity, the local best position (*Pbest*) and the global best position or the leader (*Gbest*). Equations 1 and 2 are used to update the velocity and position respectively [18][19].

$$V_{i,j}(\mathrm{t}+1) = wV_{i,j}(\mathrm{t}) + r_1 c_1\left[\mathrm{Pbest}_{i,j}(\mathrm{t}) - X_{i,j}(\mathrm{t})\right] + r_2 c_2\left[\mathrm{Gbest}(\mathrm{t}) - X_{i,j}(\mathrm{t})\right](1)$$

$$X_{i,j}(\mathrm{t}+1) = X_{i,j}(\mathrm{t}) + V_{i,j}(\mathrm{t}+1)(2)$$

where $\boldsymbol{V}_{\boldsymbol{i},\boldsymbol{j}}$ is a velocity of $\boldsymbol{i}$ particle at iteration $\boldsymbol{t}$; $\boldsymbol{X}_{\boldsymbol{i},\boldsymbol{j}}$ is a position of $\boldsymbol{i}$ particle at iteration $\boldsymbol{t}$ and it depends on previous position and new velocity, $\boldsymbol{w}$ is the inertia weight that is used to control the influence of the previous velocities on the current velocity [20], $\boldsymbol{r_1}$ and $\boldsymbol{r_2}$ are two random numbers between (0,1), $\boldsymbol{c_1}$ and $\boldsymbol{c_2}$ are learning factors or acceleration factors that are fixed numbers, $\mathbf{Pbest}_{\boldsymbol{i},\boldsymbol{j}}(\mathbf{t})$ is the local best particle $\boldsymbol{i}$ that have the smallest fitness value obtained so far in one iteration $\boldsymbol{t}$; $\boldsymbol{Gbest}(\boldsymbol{t})$ is the particle leader or global best position at generation $\boldsymbol{t}$.

The leader particle in each generation guides other particles to move towards the optimal positions. The performance of each particle in the swarm is evaluated according to objective function or the fitness function of the optimization problem [21][22].

It is assumed that a *j*-dimensions in search space and particles $\boldsymbol{i}$ (potential solutions) has a fitness value *F(x)* and a velocity *V* that makes it move in the search space. The process steps of PSO algorithm are shown in the following [23][24]:

Step 1: Initialize a random population (positions *X* and velocities *V* of all particles).
Step 2: Assume the local best particles set equals to the positions set such as: $Pbest\ i,j = X\ i,j$ and evaluate the fitness value of each particle $F(x)i,j$ (the fitness value measured in different ways according to problem) and then take the best value (either maximum or minimum) from this set to be the global best position (*Gbest*) called the leader.
Step 3: Update the particle's velocity according to equation (1) and then
Update the particle's position according to equation (2).
Step 4: Evaluate the fitness value of each particle with the new position.
Step 5: Compare the current fitness value with the previous position, if the current is better, then $Pbest\ i,j = F(x)i,j$
*Else*
$Pbest\ i,j\ (t+1) = Pbest\ i,j\ (t)$.
Step 6: If *Pbest(t+1)* is better than *Gbest(t) then*


Received: 15 May 2018/Accepted: 20 June 2018

*Gbest(t+1) = Pbest(t+1)*
Else
*Gbest(t+1) = Gbest(t).*
Step 7: If current number of iterations is larger than the maximum number of iterations then stop and return the solution, else go to step 3.

## IV. Multiple Traveling Salesman Problem

The Multiple Traveling Salesman Problem (MTSP) is a variation of the Travelling Salesman Problem (TSP). The difference between MTSP and TSP is that in MTSP there are *m* salesmen, every depot (city) in a given group of *n* cities is divided into *m* tours by assigning every of these depots (cities) to a different salesman. The objective is to seek out the minimum cost of the tours in total. The cost can be referred as distance or time [25].

The MTSP is outlined on a graph G = (V, A), where A represents the set of edges and V referred the set of vertices. Let *C = (cij)* be the cost matrix defined on the group of A. If *cij = cji* then the cost matrix is symmetric, otherwise it is asymmetric. If the cost matrix satisfies cij≤ cik + ckj for *i, j, k*, then the matrix C satisfies the triangle inequality [26][27].

Among the proposed models for the MTSP within the literature, assignment based mathematical model, therefore tree based mathematical model and a three-index row-based model have been common used [28].

The three-index row-based model for the MTSP is as follows: Let *n* be the number of cities to be visited, and *m* be the number of salesmen (we assume n ≥ 3m+1). Then the variable *xij* is defined as follows [28]:

$$x_{ij} = \begin{cases} 1, & if\ edge(i,j) is\ used\ in\ tour, \\ 0, & otherwise. \end{cases}$$

Goal function:

$$minimize \sum_{(i,j)\in A} c_{ij}\, x_{ij} \qquad (3)$$

Constraints:

$$\sum_{j=2}^{n} x_{1j} = m \qquad (4)$$
$$\sum_{j=2}^{n} x_{j1} = m \qquad (5)$$
$$\sum_{i=1}^{n} x_{ij} = 1, j = 2, \dots, n \qquad (6)$$
$$\sum_{j=1}^{n} x_{ij} = 1, i = 2, \dots, n \qquad (7)$$
$$\sum_{i\in S} \sum_{j\in S} x_{ij} \leq |S| -1, S\ V-1, S \neq 0 \qquad (8)$$
$$x_{ij} = 0 \vee 1, (i,j) \in A \qquad (9)$$

In this model, constraints (6), (7) and (8) satisfy the assignment problem constraints. Constraints (4) and (5) ensure the comeback of each salesman to their starting point. Constraint (8) is used to prevent sub-tours [29].

## V. Experimental Setup

For the purpose of investigating our hypothesis, PSO algorithm that is described in section III is implemented in order to solve the MTSP using different datasets, those datasets can be downloaded from (http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/tsp/).The total number of cities are randomly distributed among groups; the number of cities in each group is more than three cities and less than total number of cities /2.All the particles in the group have the same start and end city. The following settings are used for all the experiments:

- PSO population is 30.
- Number of Salesmen is 5.
- Number of Iteration is 500.
- Inertia weight value is 0.8.
- Velocity values are in the range [-6,6].
- MTSP datasets are Pr76, Pr152, Pr299 and Pr439.
- Number of experiments for each dataset is 5 for each learning factors value.

## VI. Results and Discussion

PSO algorithm is executed four times, each one with a dataset; the obtained results are shown in tables 1 thru 4. The values for each table is

ISSN 1943-023X

**Received: 15 May 2018/Accepted: 20 June 2018**

Table 1: Results of Pr76.

| c1 | c2 | Experiment1 | Experiment2 | Experiment3 | Experiment4 | Experiment5 | Min | Average |
|---|---|---|---|---|---|---|---|---|
| 0.5 | 0.5 | 260827 | 279298 | 262871 | 265383 | 283070 | 260827 | 270289.8 |
| 0.5 | 1 | 295216 | 287779 | 301077 | 296319 | 293417 | 287779 | 294761.6 |
| 0.5 | 1.5 | 269613 | 287071 | 301642 | 300602 | 295586 | 269613 | 290902.8 |
| 0.1 | 0.1 | 245053 | 255735 | 244447 | 237546 | 252501 | 237546 | 247056.4 |
| 1 | 1 | 285261 | 293550 | 290234 | 275592 | 287320 | 275592 | 286391.4 |
| 2 | 2 | 303282 | 293245 | 295462 | 281213 | 284266 | 281213 | 291493.6 |
| 3 | 3 | 290958 | 290232 | 269957 | 285556 | 298678 | 269957 | 287076.2 |
| 4 | 4 | 291330 | 294632 | 295884 | 292725 | 295578 | 291330 | 294029.8 |
| 0.1 | 0.5 | 269232 | 246857 | 263200 | 277621 | 279035 | 246857 | 267189 |
| 0.1 | 1 | 292862 | 286805 | 285969 | 281363 | 274708 | 274708 | 284341.4 |
| 0.1 | 1.5 | 282938 | 279327 | 283531 | 281501 | 288995 | 279327 | 283258.4 |
| 0.1 | 2 | 273676 | 252504 | 284740 | 285089 | 289609 | 252504 | 277123.6 |
| 0.1 | 2.5 | 287137 | 294241 | 282402 | 281776 | 289601 | 281776 | 287031.4 |
| 0.1 | 3 | 298487 | 278909 | 282960 | 279937 | 296257 | 278909 | 287310 |
| 0.1 | 3.5 | 294911 | 289567 | 274959 | 295764 | 287723 | 274959 | 288584.8 |
| 0.1 | 4 | 291093 | 300722 | 285270 | 304412 | 281992 | 281992 | 292697.8 |
| 0.5 | 0.1 | 224440 | 232307 | 224681 | 244014 | 242270 | 224440 | **233542.4** |
| 1 | 0.1 | 246757 | 232311 | 233193 | 236904 | 234365 | 232311 | 236706 |
| 1.5 | 0.1 | 245311 | 244852 | 240318 | 246764 | 222589 | 222589 | 239966.8 |
| 2 | 0.1 | 245429 | 240349 | 241911 | 252918 | 246917 | 240349 | 245504.8 |
| 2.5 | 0.1 | 234574 | 239267 | 242979 | 236061 | 241821 | 234574 | 238940.4 |
| 3 | 0.1 | 235888 | 237351 | 252707 | 233883 | 243097 | 233883 | 240585.2 |
| 3.5 | 0.1 | 237446 | 250389 | 241200 | 231188 | 256820 | 231188 | 243408.6 |
| 4 | 0.1 | 240967 | 217831 | 238871 | 218836 | 254645 | **217831** | 234230 |
| 0.5 | 0.5 | 260827 | 279298 | 262871 | 265383 | 283070 | 260827 | 270289.8 |

Table 2: Results of Pr152.

| c1 | c2 | Experiment1 | Experiment2 | Experiment3 | Experiment4 | Experiment5 | Min | Average |
|---|---|---|---|---|---|---|---|---|
| 0.5 | 0.5 | 423701 | 413966 | 443990 | 424888 | 434061 | 413966 | 428121.2 |
| 1 | 1 | 432201 | 409107 | 419764 | 438566 | 424918 | 409107 | 424911.2 |
| 1.5 | 1.5 | 430314 | 441548 | 447056 | 411702 | 431589 | 411702 | 432441.8 |
| 2 | 2 | 436253 | 451234 | 436193 | 450271 | 428341 | 428341 | 440458.4 |
| 2.5 | 2.5 | 453176 | 457128 | 455017 | 455231 | 456653 | 453176 | 455441 |
| 3 | 3 | 444735 | 448248 | 455977 | 448193 | 449947 | 444735 | 449420 |
| 3.5 | 3.5 | 456109 | 465903 | 443291 | 443867 | 434082 | 434082 | 448650.4 |
| 4 | 4 | 479091 | 436718 | 465811 | 474392 | 446595 | 436718 | 460521.4 |
| 0.1 | 0.1 | 361399 | 389793 | 384693 | 401416 | 362889 | 361399 | 380038 |
| 0.1 | 0.5 | 436589 | 434275 | 416377 | 435849 | 452027 | 416377 | 435023.4 |
| 0.1 | 1 | 405770 | 444468 | 444124 | 438575 | 438003 | 405770 | 434188 |
| 0.1 | 1.5 | 431624 | 449734 | 439456 | 440928 | 438195 | 431624 | 439987.4 |
| 0.1 | 2 | 433520 | 438003 | 464836 | 415045 | 425735 | 415045 | 435427.8 |
| 0.1 | 2.5 | 440430 | 440120 | 468245 | 431403 | 449121 | 431403 | 445863.8 |
| 0.1 | 3 | 458033 | 454300 | 449803 | 443909 | 449278 | 443909 | 451064.6 |
| 0.1 | 3.5 | 439999 | 434864 | 454008 | 454008 | 452686 | 434864 | 447113 |
| 0.1 | 4 | 435721 | 466299 | 435892 | 457800 | 448980 | 435721 | 448938.4 |
| 0.5 | 0.1 | 387979 | 392751 | 361645 | 384078 | 394221 | 361645 | 384134.8 |
| 1 | 0.1 | 374037 | 375733 | 331473 | 370619 | 382359 | 331473 | 366844.2 |
| 1.5 | 0.1 | 391013 | 392766 | 381083 | 394322 | 391028 | 381083 | 390042.4 |
| 2 | 0.1 | 371601 | 376201 | 381688 | 370594 | 382907 | 370594 | 376598.2 |
| 2.5 | 0.1 | 389471 | 356977 | 378839 | 387807 | 351549 | 351549 | 372928.6 |
| 3 | 0.1 | 371948 | 373678 | 382779 | 357588 | 360112 | 357588 | 369221 |
| 3.5 | 0.1 | 392825 | 376146 | 323696 | 360520 | 380511 | **323696** | **366739.6** |
| 4 | 0.1 | 351166 | 383257 | 399074 | 350562 | 380390 | 350562 | 372889.8 |

ISSN 1943-023X

**Received: 15 May 2018/Accepted: 20 June 2018**

Table 3: Results of Pr299.

| c1 | c2 | Experiment1 | Experiment2 | Experiment3 | Experiment4 | Experiment5 | Min | Average |
|---|---|---|---|---|---|---|---|---|
| 0.5 | 0.5 | 276118 | 260863 | 266901 | 269739 | 277477 | 260863 | 270219.6 |
| 1 | 1 | 275467 | 283840 | 285975 | 275407 | 285975 | 275407 | 281332.8 |
| 1.5 | 1.5 | 272859 | 277479 | 274610 | 271864 | 280109 | 271864 | 275384.2 |
| 2 | 2 | 288484 | 282434 | 292640 | 265244 | 279463 | 265244 | 281653 |
| 2.5 | 2.5 | 288096 | 282434 | 278420 | 271564 | 284711 | 271564 | 281045 |
| 3 | 3 | 290514 | 281846 | 290173 | 265071 | 284761 | 265071 | 282473 |
| 3.5 | 3.5 | 286087 | 292480 | 285594 | 284072 | 289301 | 284072 | 287506.8 |
| 4 | 4 | 270214 | 282669 | 292036 | 299057 | 288128 | 270214 | 286420.8 |
| 0.1 | 0.1 | 251891 | 260695 | 247910 | 237456 | 253309 | **237456** | 250252.2 |
| 0.1 | 0.5 | 278542 | 268080 | 269522 | 262481 | 258759 | 258759 | 267476.8 |
| 0.1 | 1 | 282375 | 274842 | 275939 | 273800 | 270957 | 270957 | 275582.6 |
| 0.1 | 1.5 | 278851 | 279277 | 283326 | 281587 | 286174 | 278851 | 281843 |
| 0.1 | 2 | 280132 | 273068 | 268620 | 278098 | 264661 | 264661 | 272915.8 |
| 0.1 | 2.5 | 273084 | 287611 | 297792 | 287309 | 276188 | 273084 | 284396.8 |
| 0.1 | 3 | 288606 | 282164 | 285607 | 294023 | 272498 | 272498 | 284579.6 |
| 0.1 | 3.5 | 291400 | 277743 | 277620 | 283504 | 274461 | 274461 | 280945.6 |
| 0.1 | 4 | 273606 | 282384 | 274589 | 282630 | 278429 | 273606 | 278327.6 |
| 0.5 | 0.1 | 248475 | 267648 | 242726 | 256840 | 240547 | 240547 | 251247.2 |
| 1 | 0.1 | 252426 | 248797 | 251207 | 249571 | 248421 | 248421 | 250084.4 |
| 1.5 | 0.1 | 253174 | 243635 | 258306 | 254861 | 257449 | 243635 | 253485 |
| 2 | 0.1 | 255079 | 254205 | 251472 | 253848 | 248869 | 248869 | 252694.6 |
| 2.5 | 0.1 | 256164 | 259166 | 254863 | 253064 | 243562 | 243562 | 253363.8 |
| 3 | 0.1 | 238266 | 256857 | 246677 | 245178 | 257281 | 238266 | **248851.8** |
| 3.5 | 0.1 | 264195 | 262273 | 257175 | 264861 | 244477 | 244477 | 258596.2 |
| 4 | 0.1 | 248948 | 256807 | 254860 | 249581 | 246985 | 246985 | 251436.2 |

Table 4: Results of Pr439.

| c1 | c2 | Experiment1 | Experiment2 | Experiment3 | Experiment4 | Experiment5 | Min | Average |
|---|---|---|---|---|---|---|---|---|
| 0.5 | 0.5 | 721391 | 733768 | 699029 | 698286 | 712214 | 698286 | 712937.6 |
| 1 | 1 | 751439 | 721865 | 715257 | 728167 | 719793 | 715257 | 727304.2 |
| 1.5 | 1.5 | 733236 | 720692 | 741865 | 737615 | 734599 | 720692 | 733601.4 |
| 2 | 2 | 740163 | 713860 | 728987 | 714920 | 737974 | 713860 | 727180.8 |
| 2.5 | 2.5 | 740940 | 758284 | 733230 | 758005 | 728082 | 728082 | 743708.2 |
| 3 | 3 | 731050 | 740125 | 755457 | 739346 | 752685 | 731050 | 743732.6 |
| 3.5 | 3.5 | 738115 | 731132 | 751683 | 748155 | 747685 | 731132 | 743354 |
| 4 | 4 | 754337 | 754337 | 747045 | 728554 | 766588 | 728554 | 750172.2 |
| 0.1 | 0.1 | 666807 | 650325 | 673785 | 676923 | 671880 | 650325 | 667944 |
| 0.1 | 0.5 | 695233 | 708297 | 684157 | 727938 | 709905 | 684157 | 705106 |
| 0.1 | 1 | 728326 | 721911 | 733069 | 721935 | 714129 | 714129 | 723874 |
| 0.1 | 1.5 | 739077 | 726468 | 724886 | 729639 | 731233 | 724886 | 730260.6 |
| 0.1 | 2 | 741839 | 744669 | 733507 | 742866 | 731320 | 731320 | 738840.2 |
| 0.1 | 2.5 | 759642 | 724148 | 733117 | 743989 | 756208 | 724148 | 743420.8 |
| 0.1 | 3 | 761002 | 768932 | 756175 | 734257 | 742448 | 734257 | 752562.8 |
| 0.1 | 3.5 | 730448 | 752241 | 752180 | 753002 | 752241 | 730448 | 748022.4 |
| 0.1 | 4 | 745264 | 735735 | 750412 | 744999 | 764778 | 735735 | 748237.6 |
| 0.5 | 0.1 | 644250 | 668097 | 661381 | 658860 | 649496 | 644250 | 656416.8 |
| 1 | 0.1 | 647539 | 649022 | 644349 | 636959 | 664793 | 636959 | 648532.4 |
| 1.5 | 0.1 | 661769 | 679815 | 651355 | 642531 | 659875 | 642531 | 659069 |
| 2 | 0.1 | 675051 | 668607 | 671714 | 651675 | 652325 | 651675 | 663874.4 |
| 2.5 | 0.1 | 636674 | 653138 | 643268 | 640645 | 666329 | 636674 | **648010.8** |
| 3 | 0.1 | 656751 | 661341 | 661604 | 669651 | 654244 | 654244 | 660718.2 |
| 3.5 | 0.1 | 659318 | 661741 | 675698 | 663509 | 628010 | **628010** | 657655.2 |
| 4 | 0.1 | 665511 | 668053 | 680017 | 650186 | 669413 | 650186 | 666636 |

The obtained results show that even when applying PSO on the same problem then the performance will vary depending on the dataset and on the values of the learning factors (c1 and c2). As in table 1 it was found that the best average value is obtained when c1=0.5 and c2=0.1, while in table 2 the best average value is obtained when c1=3.5 and c2=0.1, in table 3, when c1=3 and c2=0.1, the best average value is obtained, finally in table 4, the best average value is obtained when c1=2.5 and c2=0.1.


Received: 15 May 2018/Accepted: 20 June 2018

The obtained results strongly support the aim of this paper as it was found that even for the same problem, different values for the learning factors (c1 and c2) can lead to better results. One another interesting thing to notice from the results is that for all the used datasets, c2=0.1 gave the best average. This can give an indication about fixing the value c2 and experiment only c1.

## VII. Conclusions and Future Work

In this paper, extensive experiments are done in order to show the importance of selecting the best values for the PSO learning factors, those experiments are applied on MTSP using four different datasets. Each dataset is applied to a PSO with different values of the learning factors, and is executed for five times.

The obtained results are strongly supported the aim of this paper as it was found that it is important to carefully choose the values for the learning factors in PSO even for the same problem rather than use fixed values. In addition, it was found that, for MTSP, fixing c2 value to 0.1 gave better results than varying it.

For the possible directions for future work it is recommended to do more experiments with more MTSP dataset, also, it is recommended to apply the same settings that are used in this paper to other problems.

## References


[1] J. Kennedy and R. Eberhart, “Particle Swarm Optimization,” IEEE Int. Conf., vol. 4, pp. 1942–1948, 1995.

[2] S. Abid, A. Zafar, R. Khalid, S. Javaid, and U. Qasim, “Managing Energy in Smart Homes Using Binary Particle Swarm Optimization,” Intelligent, Softw. Intensive Syst. Adv. Intell. Syst. Comput., vol. 1, pp. 189–196, 2018.

[3] Y. Wang and M. Han, “Research on Multi-Objective Multidisciplinary Design Optimization Based on Particle Swarm Optimization,” IEEE,The Second Int. Conf. Reliab. Syst. Eng. (ICRSE 2017), 2017.

[4] Kawther A., Al-Khateeb B. and Mahmood M., Application of chaos discrete particle swarm optimization algorithm on pavement maintenance scheduling problem, Cluster Computing, DOI: 10.1007/s10586-018-2239-3 (Available Online March 2018).

[5] Mohammed, M.A., Ghani, M.K.A., Hamed, R.I., Mostafa, S.A., Ibrahim, D.A., Jameel, H.K. and Alallah, A.H., 2017. Solving vehicle routing problem by using improved K-nearest neighbor algorithm for best solution. Journal of Computational Science, 21, pp.232-240.

[6] Mostafa, S.A., Mustapha, A., Mohammed, M.A., Ahmad, M.S. and Mahmoud, M.A., 2018. A fuzzy logic control in adjustable autonomy of a multi-agent system for an automated elderly movement monitoring application. International journal of medical informatics, 112, pp.173-184.

[7] Mohammed, M.A., Al-Khateeb, B., Rashid, A.N., Ibrahim, D.A., Ghani, M.K.A. and Mostafa, S.A., 2018. Neural network and multi-fractal dimension features for breast cancer classification from ultrasound images. Computers & Electrical Engineering.

[8] Ghani, M.K.A., Mohammed, M.A., Ibrahim, M.S., Mostafa, S.A. and Ibrahim, D.A., 2017. Implementing an Efficient Expert System for Services Center Managementby Fuzzy Logic Controller, Journal of Theoretical & Applied Information Technology, 95(13).

[9] C. S. Feng, S. Cong and X. Y. Feng, "A new adaptive inertia weight strategy in particle swarm optimization," 2007 IEEE Congress on Evolutionary Computation, Singapore, 2007, pp. 4186-4190.

[10] J. C. Bansal, P. K. Singh, M. Saraswat, A. Verma, S. S. Jadon and A. Abraham, "Inertia Weight strategies in Particle Swarm Optimization," 2011 Third World Congress on Nature and Biologically Inspired Computing, Salamanca, 2011, pp. 633-640.

[11] Chauhan, P., Deep, K. & Pant, M. Memetic Comp. (2013) 5: 229. https://doi.org/10.1007/s12293-013-0111-9.

[12] Petr Maca and Pavel Pech, “The Inertia Weight Updating Strategies in Particle Swarm Optimisation Based on the Beta Distribution,” Mathematical Problems in Engineering, vol. 2015.

[13] Harrison, K.R., Engelbrecht, A.P. & Ombuki-Berman, B.M. Swarm Intell (2016) 10: 267. https://doi.org/10.1007/s11721-016-0128-z

[14] Rathore A., Sharma H. (2017) Review on Inertia Weight Strategies for Particle Swarm Optimization. In: Deep K. et al. (eds) Proceedings of Sixth International Conference on Soft Computing for Problem Solving. Advances in Intelligent Systems and Computing, vol 546. Springer, Singapore.

[15] B. Chazelle, “The Convergence of Bird Flocking,” J. ACM, vol. 61, no. 4, pp. 422–431, 2009.

[16] J. Hu, X. Zeng, and J. Xiao, “Artificial Fish School Algorithm For Function Optimization,” pp. 1–4, 2010.

[17] B. Alatas and E. Akin, “Multi-Objective Rule Mining Using a Chaotic Particle Swarm Optimization Algorithm,” Knowledge-Based Syst., vol. 22, no. 6, pp. 455–460, 2009.

[18] C. a. Coello Coello and M. Reyes-Sierra, “Multi-Objective Particle Swarm Optimizers: A Survey of the State-of-the-Art,” Int. J. Comput. Intell. Res., vol. 2, no. 3, pp. 287–308, 2006.

[19] A. U. Sikder, “Review on Single & Multi-Objective Optimization Techniques.”, 2008.


ISSN 1943-023X

**Received: 15 May 2018/Accepted: 20 June 2018**

[20] M. Onyango, S. A. Merabti, J. Owino, I. Fomunung, and W. Wu, “Analysis of Cost Effective Pavement Treatment and Budget Optimization for Arterial Roads in the City of Chattanooga,” Front. Struct. Civ. Eng., pp. 1–9, 2017.
[21] G. C. Xuncai Zhang, Xiaoxiao Wang, Ying Niu, “Chaos Multi-Objective Particle Swarm Optimization Based on Efficient Non-Dominated Sorting,” Commun. Comput. Inf. Sci., vol. 562, pp. 683–695, 2015.
[22] X. Zhang, X. Wang, Y. Niu, and G. Cui, “Improved Chaos Multi-Objective Particle Swarm Optimization,” J. Comput. Theor. Nanosci., vol. 13, no. 6, pp. 3659–3666, 2016.
[23] S. Lalwani, S. Singhal, R. Kumar, and N. Gupta, “a Comprehensive Survey: Applications of Multi-Objective Particle Swarm Optimization (MOPSO) Algorithm,” Trans. Comb. ISSN, vol. 2, no. 1, pp. 2251–8657, 2013.
[24] P. D. Dangewar, Bhagyashri D.Dipti D. Patil, “Multi-Objective Particle Swarm Optimization (MOPSO) based on Pareto Dominance Approach,” Int. J. Comput. Appl., vol. 25, no. 5, pp. 1025–1050, 2014.
[25] A. A. R. Hosseinabadi, M. Kardgar, M. Shojafar, and S. Member, “GELS-GA : Hybrid Metaheuristic Algorithm for Solving Multiple Travelling Salesman Problem GELS-GA : Hybrid Metaheuristic Algorithm for Solving Multiple Travelling Salesman Problem,” no. September, 2016.
[26] A. Singh, “A Review on Algorithms Used to Solve Multiple Travelling Salesman Problem,” 2016.
[27] E. Kivelevitch, K. Cohen, and M. Kumar, “A Market-Based Solution to the Multiple Traveling Salesmen Problem A Market-based Solution to the Multiple Traveling Salesmen Problem,” no. May 2014, 2013.
[28] T. Bektas, “The multiple traveling salesman problem: An overview of formulations and solution procedures,” Omega, vol. 34, no. 3, pp. 209–219, 2006.
[29] R. Matai, S. Prakash Singh and M.L. Mittal, “Traveling Salesman Problem: an Overview of Applications Formulations, and Solution Approaches”, Prof. Donald Davendra (Ed.), ISBN: 978-953-307-426-9.


**Received: 15 May 2018/Accepted: 20 June 2018**