IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received November 19, 2021, accepted December 2, 2021, date of publication December 6, 2021, date of current version December 21, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3133493

An Enhanced Swap Sequence-Based Particle Swarm Optimization Algorithm to Solve TSP

BIBI AAMIRAH SHAFAA EMAMBOCUS[1], MUHAMMED BASHEER JASSER[1], (Member, IEEE), MUZAFFAR HAMZAH[2], AIDA MUSTAPHA[3], (Member, IEEE), AND ANGELA AMPHAWAN[1], (Member, IEEE)

[1]Department of Computing and Information Systems, School of Engineering and Technology, Sunway University, Petaling Jaya, Selangor 47500, Malaysia
[2]Faculty of Computing and Informatics, Universiti Malaysia Sabah, Kota Kinabalu 88400, Malaysia
[3]Department of Mathematics and Statistics, Faculty of Applied Sciences and Technology, Universiti Tun Hussein Onn Malaysia, Johor 86400, Malaysia

Corresponding author: Muhammed Basheer Jasser (basheerj@sunway.edu.my)

This work was supported by the Sunway University Publication Support Scheme.

ABSTRACT The Traveling Salesman Problem (TSP) is a combinatorial optimization problem that is useful in a number of applications. Since there is no known polynomial-time algorithm for solving large scale TSP, metaheuristic algorithms such as Ant Colony Optimization (ACO), Bee Colony Optimization (BCO), and Particle Swarm Optimization (PSO) have been widely used to solve TSP problems through their high quality solutions. Several variants of PSO have been proposed for solving discrete optimization problems like TSP. Among them, the basic algorithm is the Swap Sequence based PSO (SSPSO), however, it does not perform well in providing high quality solutions. To improve the performance of the swap sequence based PSO, this paper introduces an Enhanced Swap Sequence based PSO (Enhanced SSPSO) algorithm by integrating the strategies of the Expanded PSO (XPSO) in the swap sequence based PSO. This is because although XPSO is only suitable for solving continuous optimization problems, it has a high performance among the variants of PSO. In our work, the TSP problem is used to model a package delivery system in the Kuala Lumpur area. The problem set consists of 50 locations in Kuala Lumpur. Our aim is to find the shortest route in the delivery system by using the enhanced swap sequence based PSO. We evaluate the algorithm in terms of effectiveness and efficiency while solving TSP. To evaluate the proposed algorithm, the solutions to the TSP problem obtained from the proposed algorithm and swap sequence based PSO are compared in terms of the best solution, mean solution, and time taken to converge to the optimal solution. The proposed algorithm is found to provide better solutions with shorter paths when applied to TSP as compared to swap sequence based PSO. However, the swap sequence based PSO is found to converge faster than the proposed algorithm when applied to TSP.

INDEX TERMS Metaheuristics, optimization problem, particle swarm optimization, swarm intelligence, traveling salesman problem.

I. INTRODUCTION

The Traveling Salesman problem (TSP) is a combinatorial optimization problem which consists of finding the minimum distance tour of n cities, starting and ending at the same city and visiting each city exactly once. TSP can be applied in multitudinous areas such as Mathematics, Computer Science, Operation Research, Genetics, Engineering, and Electronics [1]. Examples of its application include the machine scheduling problem, cellular manufacturing, arc routing problem, frequency assignment problem, and matrices structuring [1]. It has also been applied in X-Ray crystallography, computer wiring, and vehicle routing [2].

TSP is an NP-complete combinatorial problem, whereby large-scale TSP cannot be solved using any polynomial-time algorithm, hence there is a need for heuristic and metaheuristic algorithms to solve the problem though the solutions may not necessarily be optimal [3]. These algorithms aim at finding a suitable solution in a polynomial time by using local searches in the problem search space and some random alterations [4].

The associate editor coordinating the review of this manuscript and approving it for publication was Wentao Fan.

Recently, the use of Swarm Intelligence (SI) to solve TSP has escalated tremendously. SI consists of set of algorithms that solve problems by building multi-agent systems based on the interactions of a population of biological organisms such as insects or animals. This is inspired by the fact that simple organisms are able to coordinate and execute complex tasks successfully [5]. Having a self-learning capability and being able to adapt to external variations, these algorithms are successfully applied in multifarious areas including engineering, data mining, optimization, computational intelligence, business planning, bioinformatics, and industrial applications [6]. Examples of some real-world applications are navigation control, interferometry, planetary motion sensing, micro-robot control, malignant tumor detection, and control and image processing technologies [6]. They are also used in transportation and logistics, job-shop scheduling, swarm robotics, and optimal design of energy systems [7].

A manifold of SI algorithms such as Ant Colony Optimization (ACO), Bee Colony Optimization (BCO), Particle Swarm Optimization (PSO), and their variants have been successfully used to solve TSP through high quality solutions [8]. PSO is a type of SI algorithm that is based on social interactions between entities and were first inspired by bird flocks [9]. PSO consists of a number of particles placed in the search space of a particular problem or function. Based on its current location, each particle evaluates the objective function, and its movement through the search space is determined using its own current and best locations combined with those of one or more members of the swarm. After a number of iterations, the whole swarm is likely to reach an optimum of the fitness function. PSO is used to solve a myriad of optimization problems such as constrained optimization problems, min-max problems, and multi-objective optimization problems [10]. It is used in areas such as communication and radar systems [11], machine learning [12] and cryptography [13], [14].

The central capability of PSO is the ability of particles communicating with each other in order to reach the final solution. PSO has been successfully used to solve many optimization problems and attempts to improve its performance have resulted in a considerable amount of PSO variants. These variants usually focus on the adjustment of the parameters and the learning models. However, since these improvements are overly dependent on knowledge discovery of each particle, the entire population lacks of intelligence to tackle complex situations [15]. This results in the PSO variants still prone to getting trapped in local optima, unable to achieve high accuracy, and having a slow convergence speed while solving optimization problems such as the TSP.

Xia *et al.* [15] proposed an expanded PSO (XPSO) with three strategies to enhance the performance of PSO while solving optimization problems. This algorithm has been tested on benchmark functions from the CEC'13 test suite. The strategies include learning of each particle from both the global and local exemplars, assigning a forgetting ability to particles, and adjusting the acceleration coefficients of particles based on the population's experience. According to their experiments, the XPSO algorithm is more effective in solving different types of functions and it has a higher accuracy and faster convergence speed. However, since it caters for continuous optimization problems, discrete optimization problems cannot be solved using XPSO. Considering its potential, this paper employs the strategies of XPSO in enhancing a variant of PSO which caters for discrete optimization problems such as TSP.

The original PSO algorithm and multiple of its variants including the XPSO algorithm are suitable for solving only continuous optimization problems. To cater for discrete optimization problems like TSP, several variants of PSO have been proposed, among which, the original Swap Sequence based PSO (SSPSO) algorithm [16]. However, it does not have a high performance since it produces low quality solutions [17].

In order to improve the performance of the original swap sequence based PSO, our main contribution in this paper is an Enhanced Swap Sequence based PSO (Enhanced SSPSO) algorithm in which we integrate the strategies of the Expanded PSO (XPSO) with the original swap sequence based PSO.

The enhanced swap sequence based PSO has been applied to a TSP problem which models a package delivery system in Kuala Lumpur area. The problem set consists of 50 locations in Kuala Lumpur. Our aim is to find the shortest route in the delivery system by using the enhanced swap sequence based PSO. The new algorithm is found to be more effective than the original swap sequence based PSO in providing better quality solutions. However, it is less efficient than the original swap sequence based PSO as it takes more time to converge to the global optimal solution.

The remainder of this paper proceeds as follows. Section II presents the related work of improving PSO and its variants, Section III presents and explains the Swap Sequence based PSO (SSPSO), Section IV presents and explains the Expanded PSO (XPSO), Section V presents the proposed Enhanced Swap Sequence based PSO (Enhanced SSPSO), Section VI presents the experimental results and analysis, and finally Section VII concludes with some directions of future work.

II. RELATED WORK

In this section, we present some previous works which improve the performance of PSO while solving TSP.

In [18], [19], and [20], some methods are proposed for improving the swap sequence based PSO and they are applied to TSP.

A PSO with Partial Search and Swap Operator (PSOPS) is proposed in [18]. The objective is to obtain a better solution while solving TSP using the swap operator based PSO by applying a partial search. For the PSOPS algorithm, the velocity is calculated using the same velocity update formula used in PSO, however this velocity is considered as a tentative velocity. The Swap Operators (SOs) are then applied on each particle using this tentative velocity and tour cost is calculated. The best solution among all the solutions found is then chosen to be the next solution and the final velocity is calculated. The PSOPS algorithm is applied to solve six TSP problems and the results are compared to those of Genetic Algorithm (GA), and PSO. The results show that the proposed algorithm provides better solutions that is the minimal tour cost than both GA, and PSO.

In [19], an improved enhanced self-tentative (IEST) PSO algorithm for TSP is proposed. The goal is to improve the performance of the PSO algorithm in terms of finding better solutions while solving TSP problems consisting of more than 30 cities. The IEST PSO is based on the idea of single node adjustment, enhanced from the individual self-tentative method. It introduces an additional adjustment method based on the subsequence which is carried out after the single node adjustment for each particle. The IEST PSO algorithm is tested using instances from the TSPLIB and according to the results, it has a 90% chance of finding the global optimum for TSP problems with less than 50 cities. For TSP problems with greater number of cities, the IEST PSO algorithm produced better results as compared to the enhanced self-tentative PSO algorithm.

In [20], a hybrid algorithm based on the swap sequence based PSO, ACO and K-opt operation is proposed for solving TSP. In the proposed algorithm, ACO is first used to generate an initial set of solutions which is then used to initialize the swarm for PSO. The PSO algorithm then finds the optimal TSP path using the method of swap sequence. The 3-opt operation is used whenever the position of a particle remains stagnant over a number of operations. The hybrid algorithm is tested using benchmark TSP problems from the TSPLIB and in most cases it provides either the optimal solution or solutions which are close to the optimal solution. It also achieves a higher accuracy and lower computational complexity as compared to other algorithms such as SA-ACO-PSO, HACO, and WFA with 3-opt.

In [21]–[23], and [24], the algorithms proposed, aim to improve the performance of the discrete PSO. Discrete PSO is another variant of PSO which handles discrete optimization problems like TSP.

In [21], an improved PSO algorithm which consists of self-adaptive excellence coefficients, Cauchy operator, and 3-opt, called SCLPSO is proposed to solve TSP. The aim is to overcome the problems of premature convergence and low accuracy, which is encountered by the discrete PSO algorithm when solving TSP. The contributions include making each edge of the TSP route able to adjust itself dynamically and adaptively along the optimization process by adding heuristic information in its static excellence coefficient, regulating the inertia weight using Cauchy distribution density function, and incorporating the 3-opt local search technique. The SCLPSO algorithm is applied to several classical cases from TSPLIB, which is a text-based file format for storing graphs used in studying instances of TSPs and related problems. The results show that the SCLPSO algorithm has a better performance as compared to the PSO algorithm while solving TSP.

In [22], a discrete comprehensive learning PSO algorithm, called D-CLPSO, is proposed for solving TSP. The objective is to prevent premature convergence to local optima, which is often encountered by PSO while solving TSP and to balance its intensification and diversification. The D-CLPSO algorithm uses a novel flight equation that consists of learning from the personal best of each particle and as well from features of the problem, a lazy velocity that consists of calculating the velocity in a dimension only when needed, an eager evaluation that consists of evaluating solutions immediately after the velocity component is applied, and a Metropolis acceptance criterion that is used to determine whether to accept newly produced solutions. The D-CLPSO algorithm is tested using four TSP problems from the TSPLIB and its performance is compared to other PSO based algorithms, and other swarm intelligence algorithms. The D-CLPSO algorithm is found to have a competitive performance.

A hybrid Discrete Particle Swarm Optimization (DPSO) and Brain Storm Optimization (BSO) for solving TSP, called BSO-DPSO, is presented in [23]. The objective is to combine the exchange operator from discrete PSO and the inversion idea from chromosome structure for solving TSP. In the BSO-DPSO algorithm, the creating operator of BSO is changed so as to make it suitable for discrete problems whereby the inversion mutation operator of BSO is used for local search and the random insertion of BSO is used to introduce diversity and randomness in the search process. The BSO-DPSO algorithm is tested on Oliver30, EIL51, and ST70 from TSPLIB and the results are compared to those of GA, DPSO-2opt, Chaotic DPSO and Direction-Coordinating Ant Colony Optimization (DCACO). The BSO-DPSO algorithm achieves better results than GA, DPSO-2opt, and Chaotic DPSO and it has a higher convergence rate than DPSO-2opt.

A novel discrete PSO based on dissipative structure theory, DDPSO, is proposed in [24]. The objective is to overcome the low convergence rate of discrete PSO and to prevent local optima. The inertia coefficient is set based on dissipative structure theory to increase the accuracy and convergence speed and an operation operator is used to adjust the crossover probability between particles, thereby increasing the population diversity. The DDPSO algorithm is tested using four TPS instances from TSPLIB and it is found to provide better results than Ant Colony Genetic Algorithm (ACGA), Heuristic Genetic information-based nt colony genetic algorithm (HGI-ACGA) and New Ant Colony Algorithm (NACA).

In [25]–[29], and [30], the performance of the original PSO algorithm is improved.

In [25], a hybrid algorithm that consists of a combination of the Artificial Bee Colony (ABC) algorithm and the Particle Swarm Optimization (PSO), called ABC-PSO, is proposed to solve large-scale TSP. The objective is to improve the performance of PSO by combining it with ABC in order to avoid convergence to local optima while solving large-scale TSP. The algorithm divides the population into two subgroups at each iteration and calculates the fitness of one of the subgroups using the ABC algorithm and that of the other using PSO while then determining the global optimal adaptive value by comparing the fitness of the two subgroups. The ABC-PSO algorithm is tested using the Oliver30 problem from TSPLIB in the MATLAB environment. The results of the experiment show that both the convergence speed and convergence precision of the ABC-PSO algorithm are better as compared to the ABC and PSO algorithms.

A new improved PSO algorithm, AWIPSO, is proposed in [26], which consists of a linearly descending inertia weight, adaptively controlled acceleration coefficients, and a random grouping inversion. It is applied for real-time autonomous navigation of a USV in real maritime environment. The objective is to prevent PSO from being trapped in local optima, to enhance its robustness, and to increase the accuracy of solutions. The contributions include three optimization techniques: linearly descending inertia weight, adaptively controlled acceleration coefficients, and random grouping inversion. In the AWIPSO algorithm, the inertia weight is dynamically adjusted by linearly decreasing it over iterations, the acceleration coefficients are changed linearly in each iteration and a random grouping inversion is added, whereby the swarm is divided into subgroups in which evolution takes place independently. The AWIPSO algorithm is tested using five TSP instances from TSPLIB. It is found that the algorithm produces shorter path length, gives solutions close to the known optimum, and is more robust. However, the algorithm takes more time to converge to the global optimum.

In [27], an improved PSO algorithm, OLIPSO, is proposed which consists of an inverse learning strategy used for initialization of the swarm and dynamically adjusting inertia weight and learning factor. The objective is to improve the global search ability, the convergence accuracy, and the stability of the algorithm. The OLIPSO method combines the opposition-based learning and improved particle swarm optimization in which the inverse learning strategy is applied to a single particle, the inertia weight of particles is gradually decreased during the optimization process, hence the learning factor gradually changes to increase the social ability of particles during the iterations. The OLIPSO algorithm is tested and compared to the Multiobjective PSO (MOPSO) as well as improved PSO based on CAS theory and PSO. It is reported that the proposed algorithm improves the accuracy and stability of the PSO algorithm but its performance is lower than MOPSO.

In [28], a Hybrid Improved PSO (IHPSO) algorithm which makes use of the genetic operators and a probability strategy is proposed. The objective is to reduce the complexity of PSO and to improve its accuracy while solving TSP. The probability strategy is used to better initialize the particles. Moreover, the probabilistic crossover operator is used with the Pbest particle and the deterministic crossover operator is used with the Gbest particle. A directional mutational operator is also used to increase the population diversity. The IHPSO algorithm is tested using nine TSP instances from TSPLIB and it is found to provide better solutions than the Hybrid PSO, genetic algorithm, simulated annealing and tabu search.

In [29], a multi-subdomain grouping based PSO (MSG-PSO) is proposed for solving TSP. The objective is to improve the computing efficiency of PSO while solving small and medium-sized TSP, increase the population diversity and prevent local optima. It combines the multi-subdomain grouping, the improved mutation-based PSO, and the simplified 4-opt local search strategy to improve the performance of PSO while solving TSP. The MSGPSO algorithm is tested using 10 TSP instances from TSPLIB. It is found to be effective for small and medium sized TSP problems in terms of accuracy and computing efficiency.

A hybrid algorithm called Ant supervised by Gravitational Particle Swarm Optimization with Local Search (PSOGSA-ACO-LS) which combines Gravitational PSO (GPSO) and ACO with local search mechanism is proposed in [30] for solving TSP. In the (PSOGSA-ACO-LS) method, the number of PSO particles represent the ACO instances while the solution and route length provided by the ACO algorithm are used as the fitness of the GPSO which is used to update its particles. The PSOGSA-ACO-LS is tested using eight TSP instance from the TSPLIB. It provides best known solutions for two instances, better solution than PSO-ACO-3opt for four instances and better solution than ACO-ABC for all eight instances. However, its computation cost is found to be higher than PSO.

Table 1 shows the related works in terms of the original algorithm that they improve, their objectives, the method they employ, and the algorithm proposed.

III. SWAP SEQUENCE BASED PSO

The Particle Swarm Optimization (PSO) algorithm was originally developed to solve continuous problems such as function optimization and it has been successfully used to tackle various optimization problems. However, the PSO algorithm had to be modified in order to solve discrete optimization tasks such as TSP and numerous variants of PSO have been developed to solve TSP [17]. Among them, the swap sequence based PSO [16], which is the basic and the pioneer method to solve TSP by transforming the PSO operations of function optimization to be suitable for TSP [17]. Hence, the swap sequence based PSO has been chosen to integrate the strategies of the XPSO algorithm to solve TSP in order to improve the performance of the basic PSO algorithm used for solving TSP. The swap sequence based PSO algorithm is also used to compare the performance of the new proposed algorithm applied in TSP.

The swap sequence based PSO algorithm uses the same concept of the PSO algorithm but with the feature of swap sequence. In the swap sequence based PSO, the position of a particle represents a complete TSP tour and the velocity is a sequence of swap operators which is used to change the tour towards a new tour. A swap operator contains a pair of indices which indicates that the two cities represented by the

TABLE 1. Comparison of related works.

Paper	Original algorithm to be improved	Objectives	Improvement Method	Proposed Algorithm
[18]	Swap sequence based PSO	To improve the effectiveness, that is having better solutions.	By applying a partial search method.	PSO with Partial Search (PSOPS)
[19]	Swap sequence based PSO	Finding better solutions while solving TSP.	By enhancing the individual self-tentative method using the idea of single node adjustment.	Improved Enhanced Self-Tentative (IEST) PSO
[20]	Swap sequence based PSO	To improve performance in terms of finding better solutions.	By combining ACO, swap sequence based PSO, and 3-opt operation.	hybrid algorithm based on the swap sequence based PSO, ACO and K-opt
[21]	Discrete PSO	To overcome the problems of premature convergence and low accuracy.	By the use of self-adaptive excellence coefficients, Cauchy operator, and 3-opt.	Self-adaptive excellence coefficients, Cauchy operator and 3-opt, (SCLPSO)
[22]	Discrete PSO	To prevent premature convergence to local optima.	By using a novel flight equation, lazy velocity, eager evaluation, and Metropolis acceptance criterion.	Discrete comprehensive learning particle swarm optimization (D-CLPSO)
[23]	Discrete PSO	To improve effectiveness.	By combining the exchange operator of discrete PSO and inversion idea from chromosome structure of BSO.	BSO-DPSO
[24]	Discrete PSO	To overcome low convergence rate and to prevent local optima	By setting the inertia weight based on dissipative structure theory and using operation operator to adjust the crossover probability of particles.	DDPSO
[25]	PSO	To avoid local optima problem for large-scale TSP.	By using ABC algorithm.	ABC – PSO
[26]	PSO	To prevent the problem of local optima, enhance robustness and accuracy.	By using a linearly descending inertia weight, adaptively controlled acceleration coefficients, and random grouping inversion.	AWIPSO
[27]	PSO	To improve global search ability, convergence rate and stability.	By using an inverse learning strategy and dynamically adjusting inertia weight and learning factor.	Opposition-Based Learning and Improved Particle Swarm Optimization (OLIPSO)
[28]	PSO	To reduce the complexity of PSO and to increase its accuracy while solving TSP.	Using a probability strategy for particle initialization, probabilistic crossover with Pbest, deterministic crossover with Gbest, and directional mutational crossover.	IHPSO
[29]	PSO	To improve the computing efficiency of PSO while solving small and medium-sized TSP, increase the population diversity, and prevent local optima.	By combining the multi-subdomain grouping, the improved mutation-based PSO, and the simplified 4-opt local search strategy.	MSGPSO
[30]	Gravitational PSO	To improve the performance of PSO.	By combining GPSO and ACO with local search mechanism.	PSOGSA-ACO-LS

two indices may swap in a tour when the swap operator is applied on a solution. For example, for a solution S = (1, 3, 5, 2, 4) and swap operator SO(1, 2): S′ = S + SO(1, 2) = (1, 3, 5, 2, 4) + (1, 2) = (3, 1, 5, 2, 4).

The swap sequence based PSO algorithm makes use of the same steps as the canonical PSO algorithm which includes initialization of the particles, update of the velocity and the position of the particles, update of the personal best and global best solutions and finally taking the global best solution as the outcome when the population has converged to the global optimal solution.

Fig. 1 shows the pseudocode of swap sequence based PSO and Fig. 2 shows its flowchart.

A. INITIALIZATION

Each particle in the population gets a random solution which is represented by a list of cities and a random velocity, that is a random sequence of swap operators.

B. VELOCITY UPDATE

The velocity of each particle is updated using (1).

$$V_t = V_{t-1} \oplus r_1(P_{best} \ominus X_{t-1}) \oplus r_2(G_{best} \ominus X_{t-1}) \quad (1)$$

r_1 and r_2 are random numbers between 0 and 1, P_{best} and G_{best} are the personal best solution the particle and the global best solution of the population respectively and X_{t-1} and V_{t-1} are the position of the particle and the velocity of the particle in the previous iteration respectively.

In this equation, the operator '$\oplus$' means that the two swap sequences are merged to form a new swap sequence. For example, for swap operators, $SO_1 = (1, 2)$ and $SO_2 = (2, 3)$, $SO_1 \oplus SO_2 = ((1, 2), (2, 3))$. Moreover, the subtraction of two positions, in this case $(P_{best} \ominus X_{t-1})$ and $(G_{best} \ominus X_{t-1})$, produces a velocity, that is a swap sequence. The '$\ominus$' operator indicates the subtraction of two TSP paths. For example, for positions, $X_1 = 1, 2, 3, 4$ and $X_2 = 1, 4, 3, 2$, $X_1 \ominus X_2 = SO(2, 4)$. r_1 denotes the probability that all the swap

Algorithm 1: Swap Sequence based PSO

```
for each particle do
    Set random solution and velocity;
end
for each iteration do
    for each particle do
        Update velocity using (1);
        Update position using (2);
        if cost of current position < cost of pbest solution then
            pbest solution = current position;
        end
        if cost of current position < cost of gbest solution then
            gbest solution = current position;
        end
    end
end
Sketch global best solution;
```

FIGURE 1. Swap sequence based PSO pseudocode.

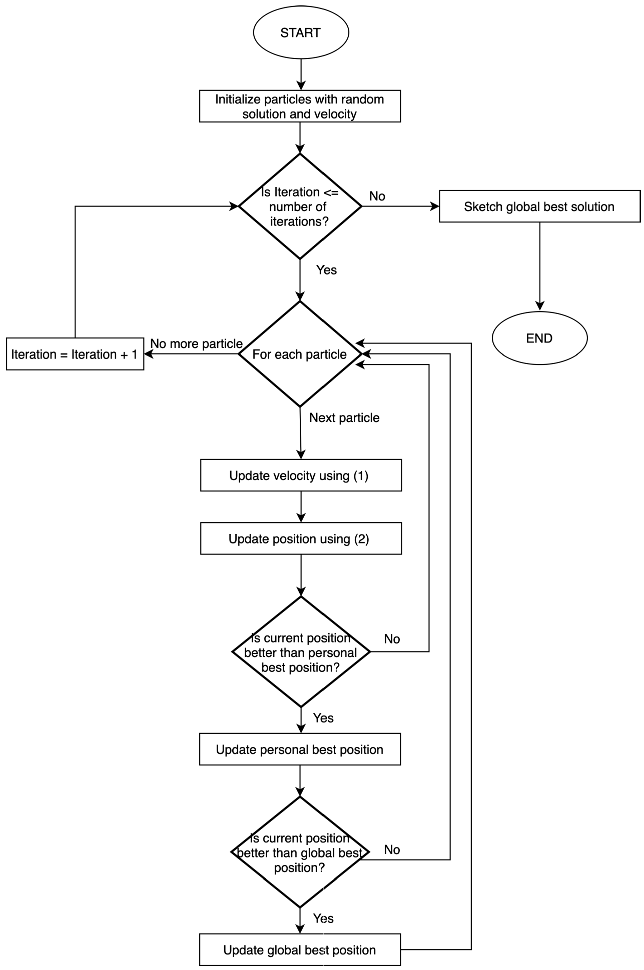


FIGURE 2. Swap sequence based PSO flowchart.

operators in the swap sequence ($P_{best} \ominus X_{t-1}$) are used and r_2 denotes the probability that all the swap operators in the swap sequence ($G_{best} \ominus X_{t-1}$) are used.

C. POSITION UPDATE

The position of the particles is then updated using (2). The '$\oplus$' operator means that the swap sequence of V_t acts on solution X_{t-1}, which is a TSP path, to get a new solution, that is a new TSP path. For example, for a TSP path, $X = 1, 2, 3, 4$ and swap sequence, $S = SO(1, 2), SO(2, 3)$, $X \oplus S = 2, 3, 1, 4$. This means that each swap operator is S is applied sequentially on X and each swap operator indicates that the positions of the two cities indicated by the swap operator will swap to produce a new TSP path.

$$X_t = X_{t-1} \oplus V_t \tag{2}$$

X_{t-1} is the position of the particle in the previous iteration and V_t is the current velocity of the particle.

D. UPDATE OF PERSONAL AND GLOBAL BEST

After updating the position of a particle using (2), the position is compared to its personal best solution and the personal best solution is updated if the current solution is better. It is also compared to the global best solution which is also updated if the current solution is better.

IV. EXPANDED (XPSO) ALGORITHM

The Expanded PSO (XPSO) algorithm is a variant of PSO which has been recently proposed by [15]. The literature has shown that XPSO generally improves the performance of the canonical PSO algorithm due to its three strategies: (1) forgetting ability assigned to particles, (2) adjustment of acceleration coefficients based on the population's experience, and (3) inclusion of both the global and local exemplars for the learning part of the particles.

The XPSO algorithm makes use of the general steps of the PSO algorithm, namely initialization, velocity update, position update, and update of the personal and global best positions.

Fig. 3 shows the pseudocode of the XPSO algorithm and Fig. 4 shows its flowchart.

The forgetting ability for each particle is first calculated before the main loop of iteration. The forgetting ability, multiple exemplars and acceleration coefficients are then included in the velocity update formula of the PSO algorithm which is then changed to the following equation (3).

$$V_t = V_{t-1} + cp.r_1\,(P_{best} - X_{t-1}) + cl.r_2((1-f_i)L_{best} - X_{t-1}) + cg.r_3((1-f_i)G_{best} - X_{t-1}) \tag{3}$$

where cp, cl and cg are three acceleration coefficients denoting the learning weights for the personal best, local best and global best exemplars respectively. r_1, $r_2$and r_3 are random numbers between 0 and 1. f_i is the forgetting ability of the particle. X_{t-1} and V_{t-1} are the position of the particle and the velocity of the particle in the previous iteration respectively.

The position update formula used is the same as the original PSO algorithm which is the following equation (4).

$$X_t = X_{t-1} + V_t \tag{4}$$

Algorithm 2: Expanded PSO (XPSO)

```
Set µ1, µ2, µ3, Stag_GB, Stag_max, p, n;
for each particle do
    Set random solution and velocity;
    Set cp, cl, cg;
    Compute forgetting ability using (5),(6);
end
for each iteration do
    for each particle do
        Update velocity using (3);
        Update position using (4);
        if cost of current position < cost of pbest solution then
            pbest solution = current position;
        end
        if cost of current position < cost of gbest solution then
            gbest solution = current position;
            Stag_GB = 0;
        else
            Stag_GB=Stag_GB+1;
            if Stag_GB ≥ Stag_max then
                for each particle do
                    Recalculate forgetting ability using (5),(6);
                    Adjust cp, cl, cg using (7);
                end
                Reselect local best topology;
            end
        end
    end
end
Sketch global best solution;
```

FIGURE 3. XPSO pseudocode.

In XPSO, a consecutive generations-stagnancy of the global best solution, $Stag_{GB}$ is used. When the global best particle is stagnant, that is it does not improve in a generation, the value of $Stag_{GB}$ is increased by 1 else it is set to 0. If $Stag_{GB}$ is larger than $Stag_{max}$, it means that the population is trapped in a deep local optimum. Therefore, the forgetting ability for each particle is recalculated, the acceleration coefficients are adjusted, and the local best topology is reselected so as to enable the population to fly out of the local optimum.

A. FORGETTING ABILITY

Forgetting is a universal biological phenomenon which balances the encoding and consolidation of new information by removing some information which are no longer useful. This forgetting ability has been beneficial for numerous algorithms solving various problems and it is believed that the different learning capabilities caused by the forgetting ability can increase genetic diversity, which will be advantageous for the evolution of species. This has inspired the inclusion of the forgetting ability to PSO.

In XPSO, the forgetting ability is included in the velocity update formula. When the velocity of a particle is updated, the particle forgets some information from its social learning exemplars based on its own characteristics. This diversifies the behavior of the population, which in turn increases its exploration. The forgetting ability depends on the familiarity of two particles. The greater the distance between two particles, the lower is the degree of familiarity and the forgetting ability should be greater. The forgetting ability of a particle depends on the distance between that particle and the global best particle. Firstly, the difference between the position of a particle and that of the global best particle is calculated based

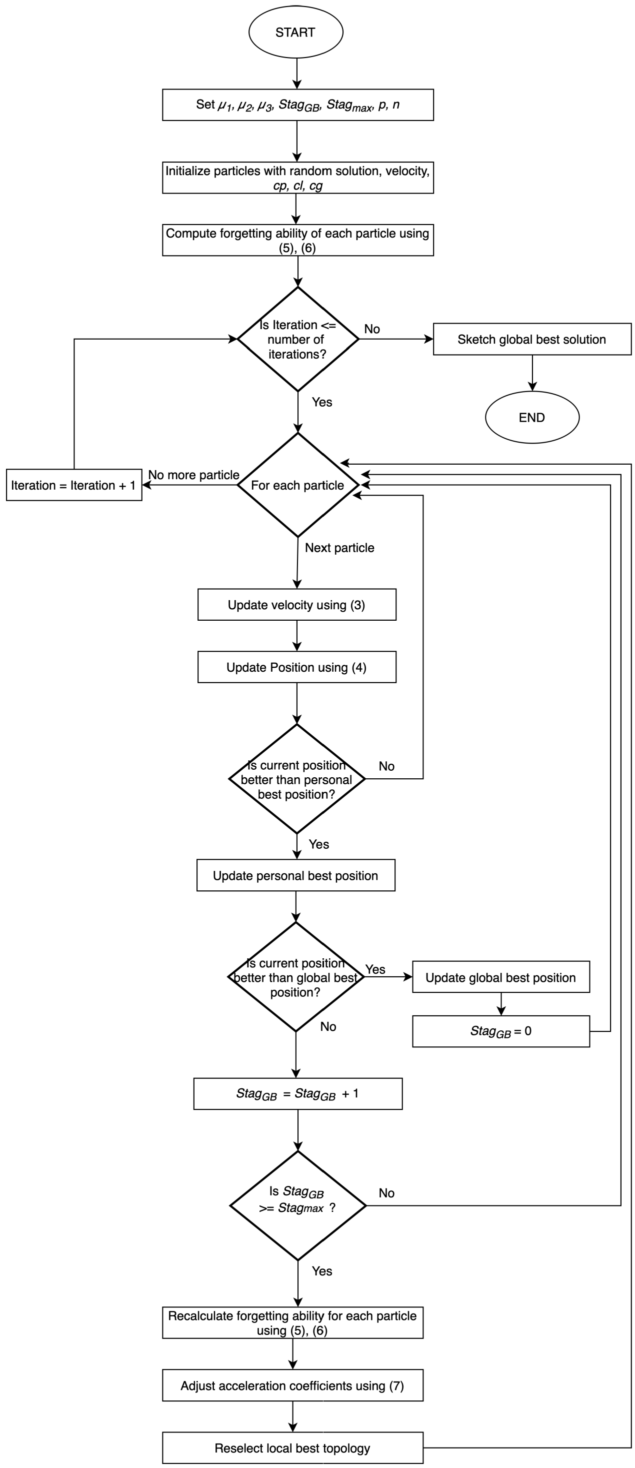


FIGURE 4. XPSO flowchart.

on the following equation (5).

$$dist_i = X_i - gb \tag{5}$$

where X_i represents the i^{th} particle and gb represents the global best particle. The particles are then sorted in ascending order by values of $dist_i$ such that the sorted population $P = X_1, X_2, \ldots X_N$ satisfies the condition $dist_1 < dist_2 < \ldots dist_N$. The forgetting ability of each particle is then calculated using equation (6).

$$f_i = \begin{cases} e^{i/N} \cdot (max(x) - min(x)) \cdot 0.05, & i > \lfloor N \cdot p \rfloor \\ 0, & otherwise \end{cases} \tag{6}$$

where $max(x)$ and $min(x)$ are the maximum and the minimum values of the entire population x. N is the total number of particles and p is a real number in the interval [0, 1].

From equation (6), it can be seen that during the initial iterations where $(max(x) - min(x))$ tends to be large, the particles have a greater forgetting ability which is beneficial for exploration while during later iterations when $(max(x) - min(x))$ is small, the forgetting ability of the particles is much less which is beneficial for exploitation. Hence, this forgetting ability balances the exploration and exploitation of the population, thereby allowing it to converge to the global optimum quickly.

B. ACCELERATION COEFFICIENTS

The XPSO algorithm uses three acceleration coefficients, cp, cl, and cg denoting the learning weights for the personal best, local best, and global best exemplar for each particle. Unlike the basic PSO algorithm, these acceleration coefficients are unique for each particle, and they are adaptively adjusted based on the historical knowledge of some elite particles in the population. In order to adjust the acceleration coefficient of a particle, three real values are randomly generated using Gaussian distribution functions $N(\mu_1, 0.1^2)$, $N(\mu_2, 0.1^2)$, and $N(\mu_3, 0.1^2)$ and are assigned to cp, cl and cg respectively. The values of μ_1, μ_2, and μ_3, are updated according to the summation of corresponding parameters of elite particles in the population. The update rule is defined as formula (7):

$$\mu_k \leftarrow (1-n) \cdot \mu_k + n \cdot avg(\langle \mu_k^{elite} \rangle), \quad 1 \le k \le 3 \tag{7}$$

where n denotes the learning weight for knowledge from elite particles and μ_k^{elite} represents a set of μ_k of the elites. n determines the degree to which μ_k of the next generation is influenced by the elites' experience. If $n = 0$, historical information of the elites is ignored and thus has no effect on the adjustment of μ_k. On the other hand, if $n = 1$, a particle ignores its own experience and the new value for μ_k depends only on the elite particles. The sum of μ_k is initially set to approximately equal to 4.0 since it is a popular rule in PSO to consider the sum of two acceleration coefficients as 4.0 [15]. Hence, at the initial evolutionary stage, $\mu_1 = \mu_2 = \mu_3 = 1.35$ is used. The acceleration coefficients are recalculated whenever the $Stag_{GB}$ becomes greater than $Stag_{max}$.

C. MULTIPLE EXEMPLARS

The learning model of the basic PSO algorithm is usually classified as either global best or local best in which a particle learns from the experience of the entire population or from its local neighbour's experience respectively. However, choosing an optimal static learning model beforehand is often difficult since the optimization process is dynamic and each of the learning model has its own advantages. Hence, in the XPSO algorithm both the global best and local best exemplars are included in the learning part of each particle so as to improve the social-learning part of the algorithm. In order to reselect the Local PSO topology for the population, the $Stag_{GB}$ is used again. When $Stag_{GB}$ becomes greater than $Stag_{max}$, the population randomly reselects a new local PSO topology. All particles in the population are assigned a random permuted order of numbers and each particle selects its left and right particles as its neighbours.

V. ENHANCED SWAP SEQUENCE BASED PSO

In this paper, the swap sequence based PSO is enhanced by using the strategies of the XPSO algorithm, namely, the forgetting ability for the particles, the adjustment of the acceleration coefficients based on the population's experience and the inclusion of both the global and local exemplars for the learning part of the particles.

The velocity and the position of the particles are updated using the method of swap sequence described in the swap sequence based PSO. In order to include the forgetting ability, multiple exemplars and acceleration coefficients in the swap sequence based PSO algorithm, the velocity update formula in (1) is changed to the following equation (8).

$$V_t = V_{t-1} \oplus cp \cdot r_1 \left(P_{best} \ominus X_{t-1}\right) \oplus cl \cdot r_2((1-f_i)L_{best} \ominus X_{t-1}) \oplus cg \cdot r_3((1-f_i)G_{best} \ominus X_{t-1}) \tag{8}$$

where cp, cl, and cg are three acceleration coefficients denoting the learning weights for the personal best, local best and global best exemplars respectively. r_1, r_2, and r_3 are random numbers between 0 and 1. P_{best}, L_{best}, and G_{best} are the particle's personal best solution, local best solution, and the global best solution of the population, respectively. Since a solution in the enhanced swap sequence based PSO algorithm is a TSP path, the P_{best}, L_{best} and G_{best} are TSP paths. f_i is the forgetting ability of the particle. X_{t-1} and V_{t-1} are the position of the particle, which is a TSP path, and the velocity of the particle, which is a sequence of swap operators, in the previous iteration, respectively.

In equation (8), the operator '$\oplus$' means that the two swap sequences are merged to form a new swap sequence similar to the velocity update formula used in the swap sequence based PSO. Also, the '$\ominus$' operator means that two TSP paths are subtracted to produce a swap sequence and hence the subtraction of two positions, in this case $(P_{best} \ominus X_{t-1})$, $(L_{best} \ominus X_{t-1})$ and $(G_{best} \ominus X_{t-1})$, produces a velocity, that is a swap sequence. The '$\oplus$' and the '$\ominus$' operators have been discussed in the swap sequence

```
Algorithm 3: Enhanced Swap Sequence based PSO
Set µ1, µ2, µ3, Stag_GB, Stag_max, p, n;
for each particle do
    Set random solution and velocity;
    Set cp, cl, cg;
    Compute forgetting ability using (5),(6);
end
for each iteration do
    for each particle do
        Update velocity using (8);
        Update position using (2);
        if cost of current position < cost of pbest solution then
            pbest solution = current position;
        end
        if cost of current position < cost of gbest solution then
            gbest solution = current position;
            Stag_GB = 0;
        else
            Stag_GB=Stag_GB+1;
            if Stag_GB ≥ Stag_max then
                for each particle do
                    Recalculate forgetting ability using (5),(6);
                    Adjust cp, cl, cg using (7);
                end
                Reselect local best topology;
            end
        end
    end
end
Sketch global best solution;
```

FIGURE 5. Enhanced SSPSO pseudocode.

based PSO section. The multiplication of a constant by a sequence of swap operators, in this case, $(1-f_i) \times L_{best}$ and $(1-f_i) \times G_{best}$, implies that the length of a swap sequence is increased if the constant is greater than one, otherwise it is truncated.

$cp.r_1$ denotes the probability that all the swap operators in the swap sequence $(P_{best} \ominus X_{t-1})$ are used, $cl.r_2$ denotes the probability that all the swap operators in the swap sequence $((1-f_i)L_{best} \ominus X_{t-1})$ are used and $cg.r_3$ denotes the probability that all the swap operators in the swap sequence $((1-f_i)G_{best} \ominus X_{t-1})$ are used. $(1-f_i)$ determines how much information is taken from $(L_{best} \ominus X_{t-1})$ and $(G_{best} \ominus X_{t-1})$.

The position update equation used is the same as in the swap sequence based PSO, that is equation (2).

Fig. 5 shows the pseudocode for the enhanced swap sequence based PSO and Fig. 6 shows its flowchart.

In the proposed enhanced swap sequence based PSO algorithm, the initialization stage is as follows: first, the values for $\mu_1, \mu_2, \mu_3, Stag_{GB}, Stag_{max}, p$, and n are set, similar to the XPSO algorithm, the particles are then initialized with a random solution, velocity, cp, cl and cg and then the forgetting ability of each particle is calculated. The random solution for the enhanced swap sequence based PSO is a complete TSP path and the random velocity is a sequence of swap operators.

In the main loop of optimization, the velocity and the position of the particles are updated using the swap sequence method according to equation (8) and (2), respectively. The

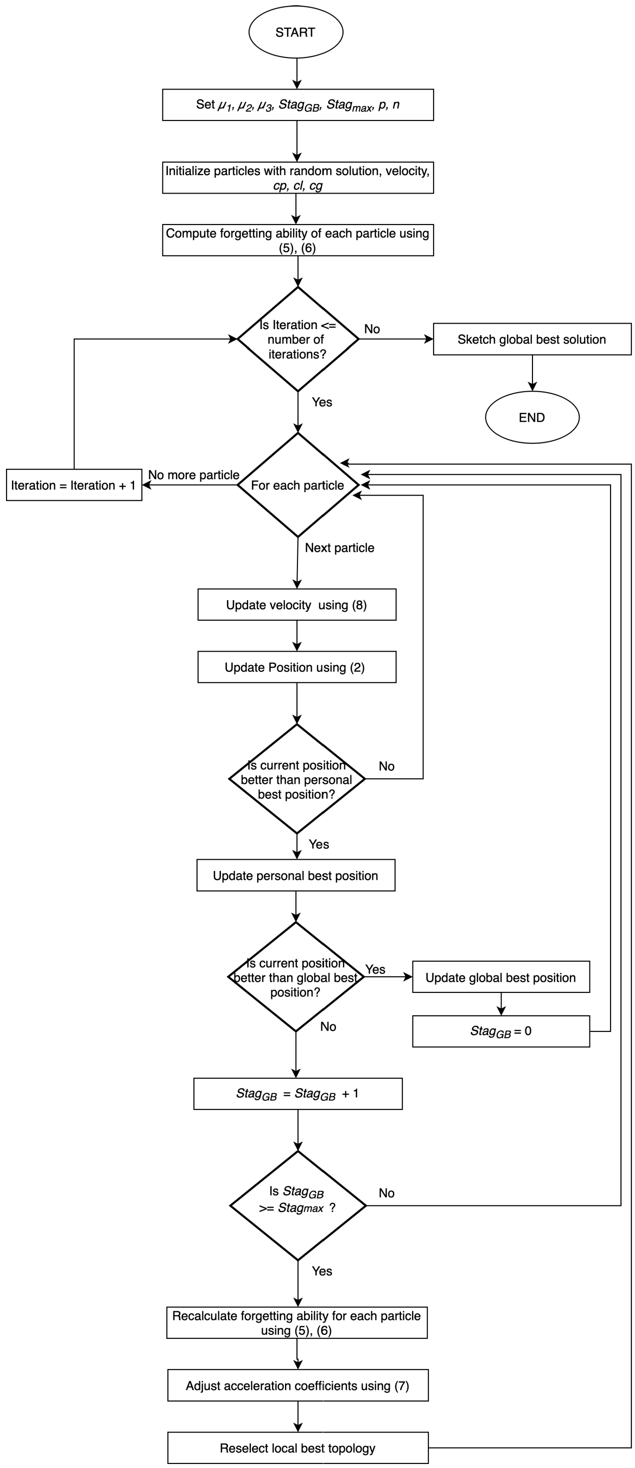


FIGURE 6. Enhanced SSPSO flowchart.

personal and global best positions are then updated by comparing the cost of the personal and global best positions to the cost of the current position and the consecutive generations-stagnancy of the global best solution, $Stag_{GB}$ of XPSO is used to determine whether the global best particle is stagnant. In the enhanced swap sequence based PSO, the

TABLE 2. Experimental parameters.

Parameter	Description	Value
X_t	The position of a particle at iteration t.	A TSP path, example: [1,3,9,6,4,2,7,8,5,10]
X_{t-1}	The position of a particle at iteration t-1.	A TSP path, example: [1,5,2,9,8,4,6,7,3,9,10]
V_t	The velocity of a particle at iteration t.	A sequence of swap operators, example: [(1,2),(3,2),(5,1)]
V_{t-1}	The velocity of a particle at iteration t-1.	A sequence of swap operators, example: [(1,2),(3,2),(5,1)]
cp	Acceleration coefficient for the personal best exemplar	In range [1.05, 1.35]
cl	Acceleration coefficient for the local best exemplar	In range [1.05, 1.35]
cg	Acceleration coefficient for the global best exemplar	In range [1.05, 1.35]
r_1	Random value	In range [0, 1]
r_2	Random value	In range [0, 1]
r_3	Random value	In range [0, 1]
P_{best}	The personal best solution of a particle	A TSP path, example: [1,5,2,9,8,4,6,7,3,9,10]
L_{best}	The local best solution of a particle	A TSP path, example: [1,5,2,9,8,4,6,7,3,9,10]
G_{best}	The global best solution of a particle	A TSP path, example: [1,5,2,9,8,4,6,7,3,9,10]
f_i	The forgetting ability for the i^{th} particle	A real value, example: 1.5
$Stag_{GB}$	The consecutive generations-stagnancy of the global best solution	In range [0,15]
$Stag_{max}$	The maximum value for the consecutive generations-stagnancy of the global best solution	15
p	Real number in interval [0,1]	0.5
μ_1	Mean value to generate cp using gaussian distribution	1.35
μ_2	Mean value to generate cl using gaussian distribution	1.35
μ_3	Mean value to generate cg using gaussian distribution	1.35
n	Learning weight for knowledge from elite particles	0.2

personal best position is the position of the particle representing the TSP path with the least cost, that is the best position that a particle has found in all the iterations and the global best position is the position of a particle in the population which represents the TSP path with the least cost among all the particles in the population. If the global best position does not improve in an iteration, the value of $Stag_{GB}$ is increased by 1 else it is set to 0 and when $Stag_{GB}$ is larger than $Stag_{max}$, the forgetting ability for each particle is recalculated, the acceleration coefficients are adjusted, and the local best topology is reselected similar to XPSO. This is to allow the population to get out of the local optimum in which it has been trapped. The local best topology of a particle consists of two neighbouring particles. This is determined by assigning a random order for all the particles in the population and then taking the right and left particles as the neighbour of a particle. The local best particle is then the one whose position represent the TSP path with the least cost among the three particles. The value of $Stag_{max}$ used for the enhanced swap sequence based PSO is 15. This value is determined by experimental results where it is found to provide the best results.

VI. EXPERIMENTAL RESULTS AND ANALYSIS

In this paper, the enhanced swap sequence based PSO algorithm is applied to solve the Traveling Salesman Problem (TSP) in a package delivery system which consists of 50 locations in Kuala Lumpur. The goal is to reduce the shipping cost for shipping companies by finding the most efficient route that can be taken by a driver who is delivering items from a shipping company. For comparison purposes, the new algorithm and the original swap sequence based PSO [16] are applied to the same TSP problem. The solutions obtained from both the new algorithm and the original swap sequence based PSO are recorded and compared in terms of best solution, mean solution, and time taken to converge to optimal solution.

A. EXPERIMENTAL SETUP

The test data representing a TSP problem that consists of 50 locations in Kuala Lumpur has been applied to both the new algorithm and swap sequence based PSO. Both algorithms were run 10 times. The number of iterations and population size used for both algorithms are 100 and 20 respectively. The solutions obtained were recorded and compared in terms of the cost of the mean solution, the best solution, and the time taken to converge to optimal solution. The solution obtained represent a complete TSP path, which in this case is the route that can be taken by a vehicle, and its cost is the total distance of the route in terms of kilometers. The distance between two locations is taken as the shortest distance that can be taken by a vehicle based on Google Maps.

The test data was then changed to represent a TSP problem consisting of 10, 20, 30, and 40 locations in Kuala Lumpur respectively and each time the test data was applied to both the new algorithm and the original swap sequence based PSO and they were run 10 times. The solutions obtained were recorded and compared in terms of the cost of mean solution, the best solution, and the time taken to converge to optimal solution. The population size for both algorithms was kept at 20. The number of iterations was then changed to 50, 200, 300, 500, and 1000 and the experiments were repeated for 10, 20, 30, 40, and 50 cities.

Both the new algorithm and the swap sequence based PSO were implemented using python programming language and all experiments were conducted on a macOS Big Sur operating system, 2.9 GHz Dual-Core Intel Core i5 CPU, and 8 GB RAM.

B. EXPERIMENTAL PARAMETERS

Table 2 shows the parameters used, their description, and their values.

For X_t, X_{t-1}, P_{best}, L_{best}, and G_{best}, the values shown are examples of a TSP path which can be obtained for the parameters. Similarly, for V_t, and V_{t-1}, examples of a swap sequence which can be obtained for the parameters are shown.

For cp, cl, and cg, the range of values which can be obtained for the parameters is shown. This range is based on the Gaussian distribution functions which is explained in section IV-B. The range of values for r_1, r_2, and r_3, is based on the original PSO algorithm.

The value of f_i shown is an example of the value which can be obtained when it is calculated using equation 6. The values for μ_1, μ_2, and μ_3, have been taken from the continuous XPSO algorithm in [15].

The values for p, n, and $Stag_{max}$ used in this paper have been determined by experimental analysis in order to get the best solutions. The range of values for $Stag_{GB}$ is then based on $Stag_{max}$.

C. RESULTS AND DISCUSSION

Table 3 shows the comparison between the performance of both the new algorithm and that of the swap sequence based PSO when applied in TSP.

Fig. 7, 8, and 9 show the convergence curve of the swap sequence based PSO and the enhanced swap sequence based PSO for 100, 300, and 500 iterations respectively. These graphs show the cost of the global best solution obtained by the swap sequence based PSO and the enhanced swap sequence based PSO at each iteration for one experiment, thereby showing how the two algorithms converge to the global best solution.

From Fig. 7, it can be seen that the enhanced SSPSO algorithm converges at 20, 30, 60, 80, and 90 iterations for 10, 20, 30, 40, and 50 locations respectively.

Fig. 10 shows the cost of the mean solutions obtained by swap sequence based PSO and enhanced swap sequence based PSO when the number of iterations for both the algorithms is changed from 50 to 1000. The mean cost is obtained by conducting each experiment 10 times and then taking the average cost obtained when the algorithms have converged.

From Fig. 10, it can be seen that the mean cost obtained by the enhanced SSPSO algorithm is always lower than that obtained by the original SSPSO algorithm, even when the maximum number of iterations is increased.

Fig. 11 shows the mean time taken by the swap sequence based PSO and enhanced swap sequence based PSO when the number of iterations is changed from 50 to 1000.

Fig. 12 shows the mean time taken by the swap sequence based PSO and enhanced swap sequence based PSO when the number of locations is changed from 10 to 50.

From the results, it can be seen that the new algorithm provides better solutions as compared to the swap sequence based PSO. In all test cases, that is a TSP problem consisting of 10, 20, 30, 40, and 50 locations respectively, the cost of the mean solution provided by the new algorithm is lower than that provided by the swap sequence based PSO. This means that the new algorithm provides shorter paths, which is better solutions when solving each TSP instance defined as compared to the swap sequence based PSO. When considering the best solution which has been obtained from each algorithm among the 10 runs, the new algorithm again provides a better solution that is a shorter path than that provided by the swap sequence based PSO in most cases. From the graphs as well, it can be seen that the enhanced swap sequence based PSO converge to solutions with lower costs as compared to the swap sequence based PSO.

In terms of efficiency, the swap sequence based PSO is found to be better than the new algorithm when applied to TSP as the swap sequence based PSO algorithm takes less time to converge to optimal solution for the TSP problems defined.

Hence, it can be deduced that the new algorithm provides better solutions than the swap sequence based PSO when applied to TSP as the paths provided by the new algorithm is shorter than those provided by the swap sequence based PSO. However, the swap sequence based PSO is found to be more efficient than the new algorithm applied in TSP as the swap sequence based PSO takes less time to converge to its optimal solution.

In the case of package delivery systems, the enhanced swap sequence based PSO algorithm will be more advantageous than the original SSPSO even if the enhanced SSPSO does not provide a huge improvement in terms of the cost. This is because the cost obtained shows the distance in kilometers and reducing the distance that a vehicle undertakes on a daily basis even by a few kilometers will considerably reduce the fuel cost and the time taken to deliver the packages in the long term. Moreover, even if the enhanced SSPSO algorithm takes more time to obtain the optimal solution, this will not be a major problem if the delivery system configures the route before starting the journey.

Fig. 13 shows an example of the path provided by the enhanced swap sequence based PSO algorithm when it is applied to solve a TSP problem consisting of 20 locations in Kuala Lumpur. For simplicity purposes, the path provided for only 20 locations is shown. The Figure was made using Google Maps. The path provided is as follows and the total cost of this path is found to be 147:

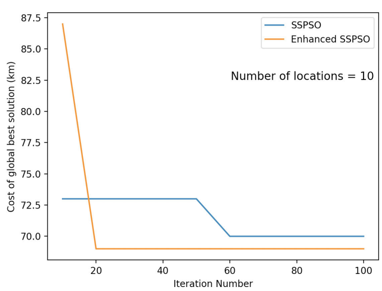
$1 \rightarrow 13 \rightarrow 19 \rightarrow 5 \rightarrow 6 \rightarrow 14 \rightarrow 8 \rightarrow 3 \rightarrow 10 \rightarrow 9 \rightarrow 7 \rightarrow 17 \rightarrow 20 \rightarrow 4 \rightarrow 12 \rightarrow 2 \rightarrow 15 \rightarrow 18 \rightarrow 16 \rightarrow 11 \rightarrow 1$.

D. TECHNICAL ANALYSIS OF ALGORITHM

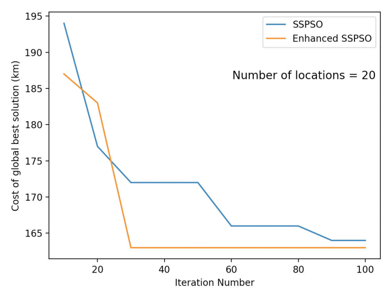
In this section, We present a technical justification about the performance of the proposed algorithm by discussing the effect of using the forgetting ability, the adaptive acceleration coefficients, and the multiple exemplars.

TABLE 3. Comparison of performance of proposed algorithm and swap sequence based PSO applied in TSP.

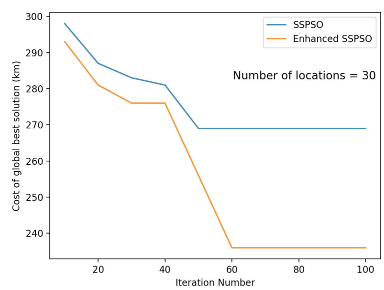
		Swap sequence based PSO			Enhanced Swap sequence based PSO		
Number of locations	Number of iterations	Cost of Mean Solution (km)	Cost of Best Solution (km)	Mean Time Taken (s)	Cost of Mean Solution (km)	Cost of Best Solution (km)	Mean Time Taken (s)
10	50	72.0	66.0	0.0761	70.1	66.0	0.116
10	100	69.3	64.0	0.265	68.4	64.0	0.733
10	200	69.5	65.0	1.251	67.4	63.0	3.501
10	300	69.2	64.0	2.303	67.9	63.0	7.757
10	500	66.1	64.0	8.784	65.4	63.0	27.717
10	1000	66.4	64.0	32.887	65.0	63.0	92.901
20	50	167.0	154.0	0.116	166.1	150.0	0.415
20	100	166.1	158.0	0.658	160.4	145.0	2.614
20	200	161.7	151.0	3.682	159.3	151.0	13.223
20	300	160.6	153.0	4.793	158.5	146.0	33.161
20	500	157.8	145.0	17.488	155.5	147.0	177.758
20	1000	151.9	143.0	93.275	149.4	125.0	818.142
30	50	266.8	250.0	0.202	264.2	248.0	0.908
30	100	258.2	242.0	1.062	257.8	249.0	5.137
30	200	254.8	241.0	5.981	254.2	244.0	26.636
30	300	255.9	246.0	11.414	250.0	230.0	66.412
30	500	252.1	242.0	19.433	251.1	242.0	223.909
30	1000	247.0	239.0	119.626	244.2	230.0	1056.296
40	50	426.5	403.0	0.321	419.2	396.0	1.841
40	100	415.6	404.0	1.792	415.1	399.0	9.735
40	200	413.0	407.0	6.540	403.8	391.0	52.392
40	300	409.5	388.0	18.745	408.5	385.0	135.628
40	500	408.7	391.0	45.209	401.5	389.0	317.081
40	1000	398.5	383.0	227.144	395.1	381.0	1930.985
50	50	519.8	512.0	0.368	515.7	488.0	2.560
50	100	508.4	483.0	1.913	501.1	479.0	13.163
50	200	506.6	494.0	7.711	500.0	484.0	61.592
50	300	502.6	495.0	16.929	498.6	480.0	156.847
50	500	498.8	485.0	66.572	495.9	478.0	446.417
50	1000	494.4	474.0	263.912	490.6	476.0	1960.538



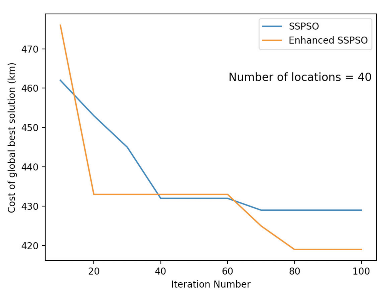
(a) 10 locations.



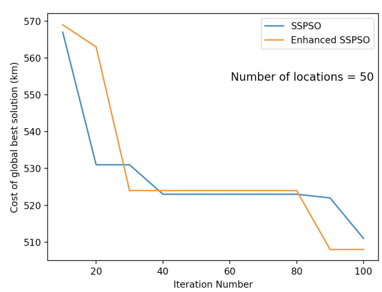
(b) 20 locations.



(c) 30 locations.



(d) 40 locations.

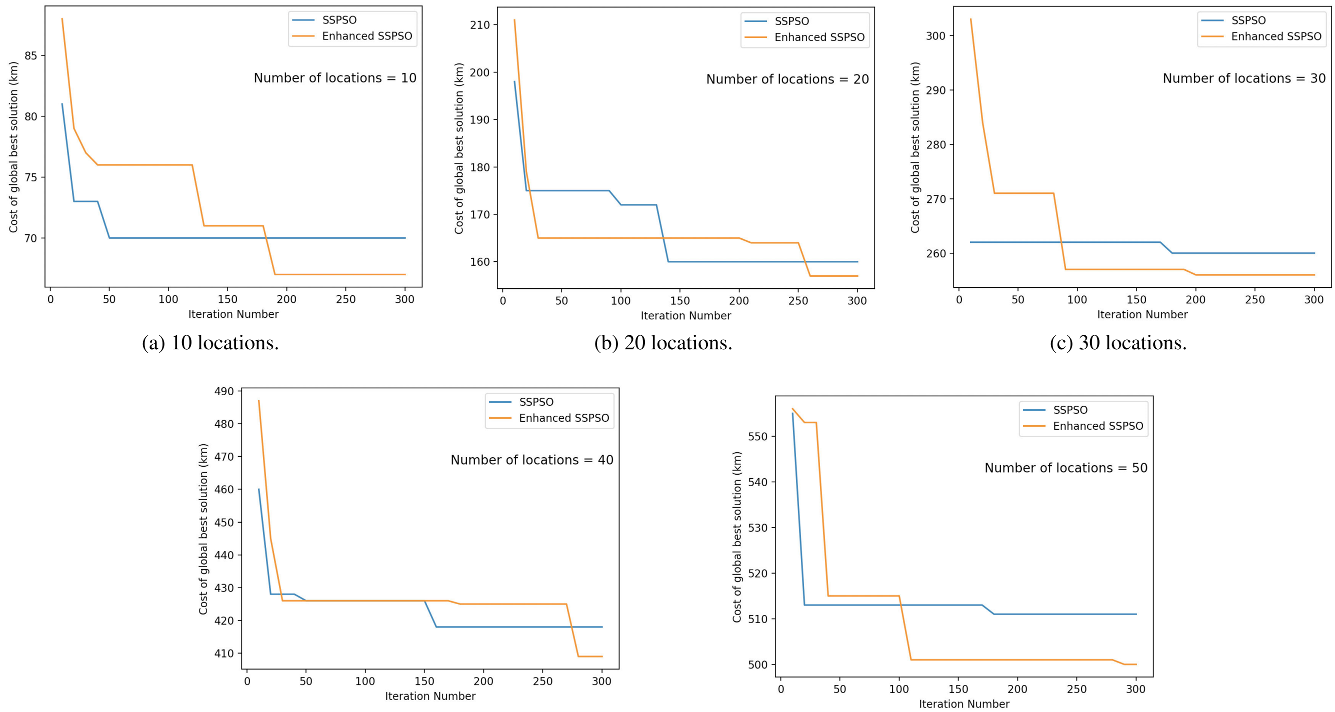


(e) 50 locations.

FIGURE 7. Performance of SSPSO and Enhanced SSPSO for 100 iterations.

1) EFFECT OF FORGETTING ABILITY

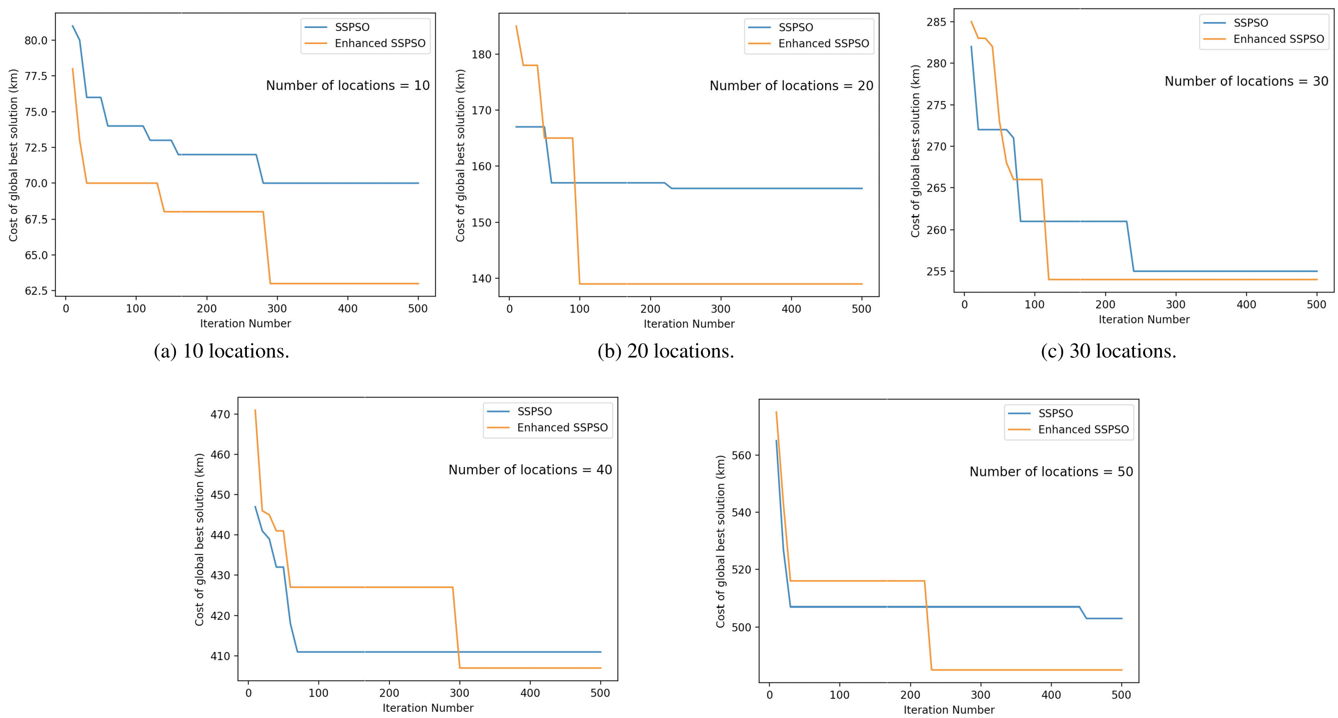
The original PSO algorithm and also its variants make use of historical information which is a crucial part of the algorithm. This is usually in the form of the personal best solution which is the best solution which has been found by a particle and the global best solution which is the best solution which



(a) 10 locations. (b) 20 locations. (c) 30 locations.

(d) 40 locations. (e) 50 locations.

FIGURE 8. **Performance of SSPSO and Enhanced SSPSO for 300 iterations.**

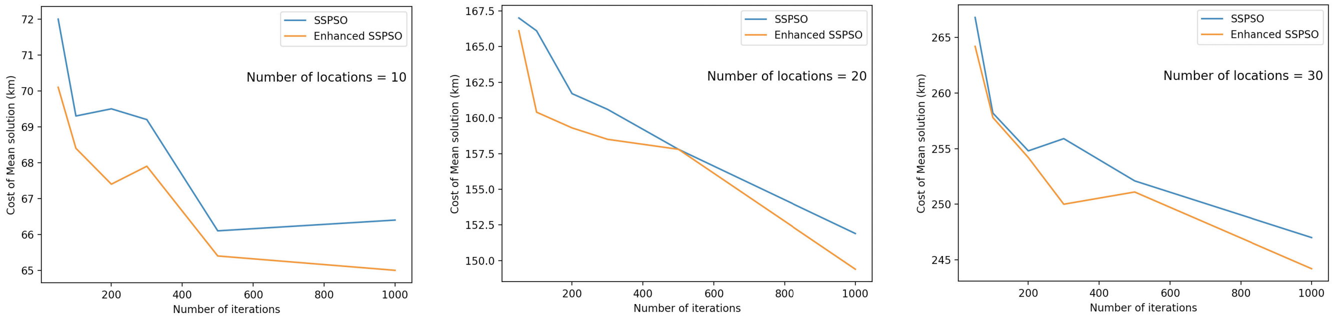


(a) 10 locations. (b) 20 locations. (c) 30 locations.

(d) 40 locations. (e) 50 locations.

FIGURE 9. **Performance of SSPSO and Enhanced SSPSO for 500 iteration.**

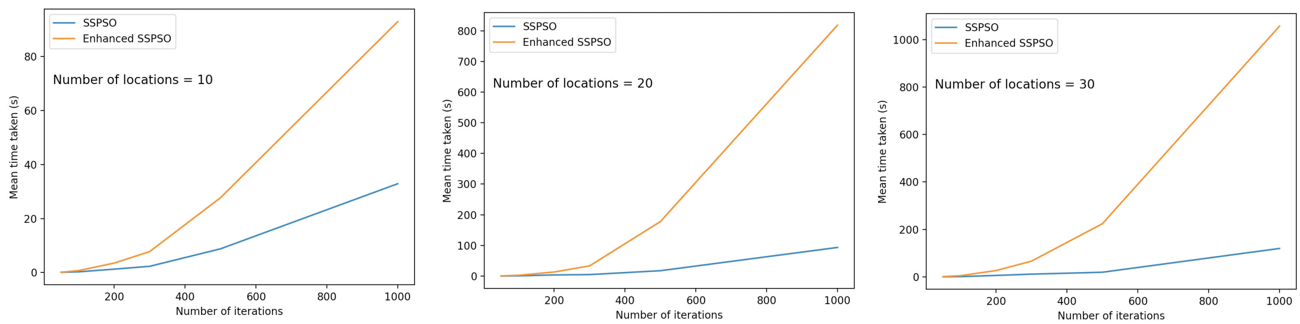
has been found by the entire population. However, some of these information might not be beneficial for updating the position of a particle at some point. The forgetting ability of the particles allow them to neglect some information which are not beneficial while updating their positions. In terms of the enhanced swap sequence based PSO applied to TSP,



(a) 10 locations. (b) 20 locations. (c) 30 locations.

(d) 40 locations. (e) 50 locations.

FIGURE 10. Cost of mean solutions given by SSPSO and Enhanced SSPSO.



(a) 10 locations. (b) 20 locations. (c) 30 locations.

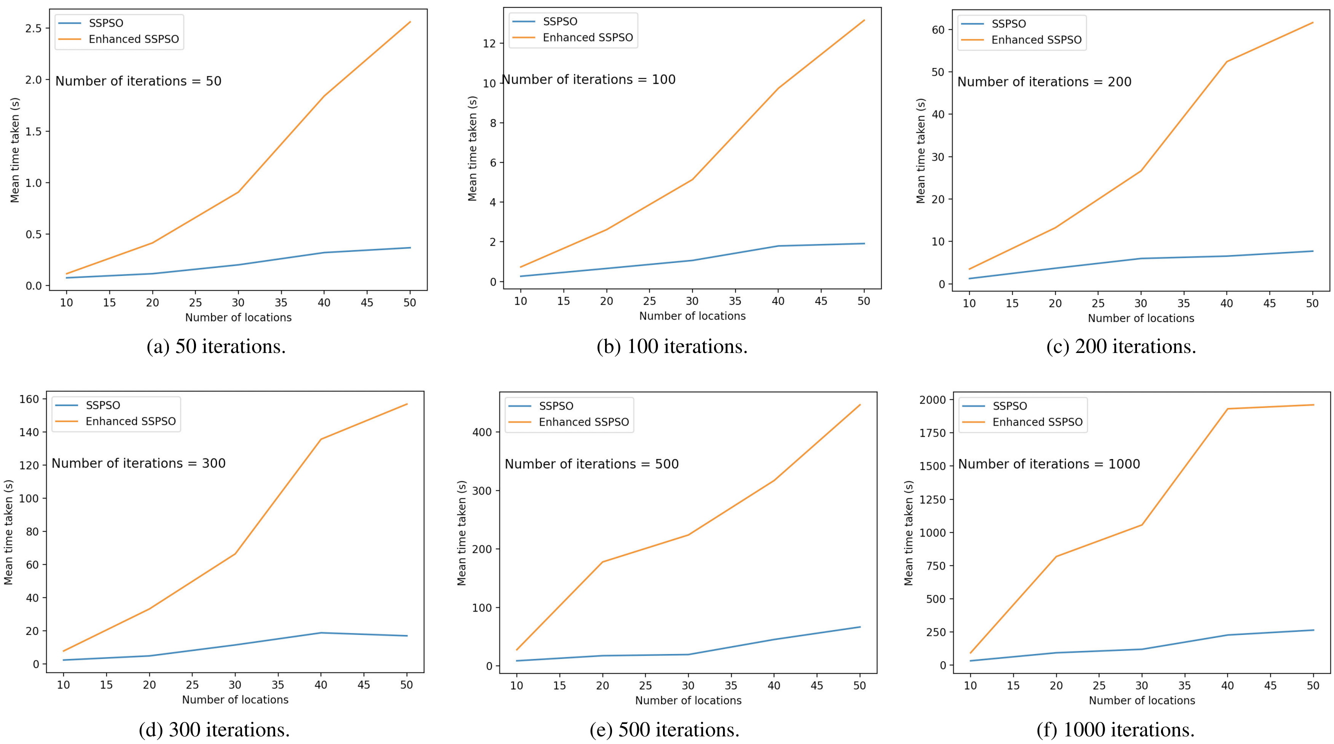
(d) 40 locations. (e) 50 locations.

FIGURE 11. Mean time taken by SSPSO and Enhanced SSPSO.

the forgetting ability allows the particles to neglect some information from the local and global best exemplars. Hence, while updating its position, a particle focuses on the beneficial historical information from the local and global best exemplars.

2) EFFECT OF ADAPTIVE ACCELERATION COEFFICIENTS

The enhanced swap sequence based PSO algorithm makes use of acceleration coefficients which are adaptively tuned so that they always have a favourable effect. Also, unlike in the original PSO, each particle has its own acceleration



(a) 50 iterations. (b) 100 iterations. (c) 200 iterations.

(d) 300 iterations. (e) 500 iterations. (f) 1000 iterations.

FIGURE 12. Mean time taken by SSPSO and Enhanced SSPSO.

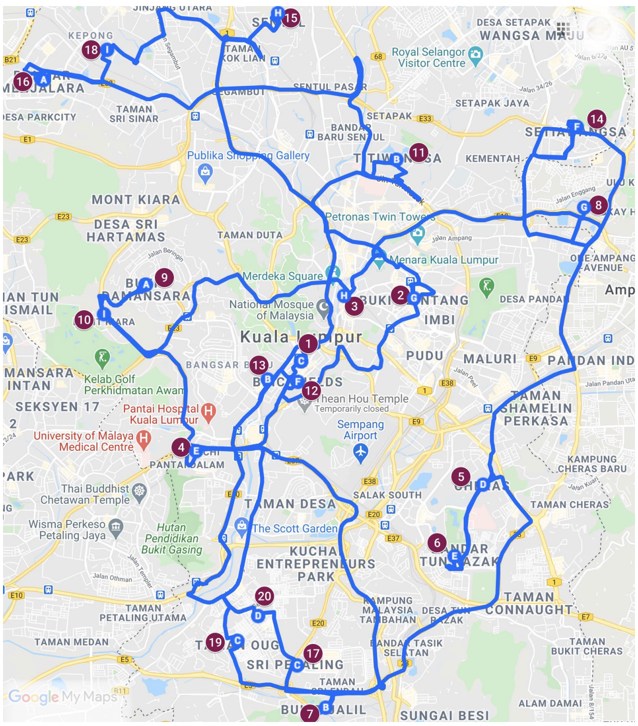


FIGURE 13. TSP path provided by enhanced SSPSO algorithm for 20 cities.

coefficient. These acceleration coefficients are adjusted based on some elite particles from the population and hence the useful knowledge of these particles are used when a particle is updating its position. The acceleration coefficients are adjusted whenever the value of $Stag_{GB}$ becomes greater than the value of $Stag_{max}$. This means that the global best solution is not improving over a certain amount of iterations and the acceleration coefficients are no longer effective. Hence, by using the adaptive acceleration coefficients, it is ensured that they always have a favourable effect in updating the position of a particle.

3) EFFECT OF MULTIPLE EXEMPLARS

The original PSO algorithm and numerous of its variants usually make use of only one exemplar in addition to the personal best exemplar; either the local best exemplar in which the neighbourhood consists of some particles from the population and the global best solution is determined based on these particles or the global best exemplar in which the neighbourhood consists of the whole population and the global best solution is determined based on the whole population. Since both of these exemplars have their own advantages, it is difficult to determine which one will be the optimal one for a specific problem [15]. Hence, by employing both of them, the advantages of learning from both the local and global exemplars are exploited.

VII. CONCLUSION AND FUTURE WORK

The traveling salesman problem is a discrete optimization problem which is usually solved using heuristic and meta-heuristic algorithms since large-scale TSP cannot be solved using any polynomial-time algorithm. A number of swarm intelligence algorithms and their variants including those of the particle swarm optimization have been used to solve

TSP by providing high quality solutions. Since the original PSO algorithm caters for continuous optimization problems, several of its variants have been proposed to cater for discrete optimization problems like TSP, among which the swap sequence based PSO is the pioneer one. However, it does not have a high performance since it produces low quality solutions.

In this paper, a new enhanced swap sequence based PSO algorithm is proposed which uses the strategies of the XPSO algorithm to improve the original swap sequence based PSO. This is because although XPSO is only suitable for solving continuous optimization problems, it has a high performance among the variants of PSO. We have demonstrated the strategies of the Expanded Particle Swarm Optimization (XPSO) algorithm which includes learning of each particle from both the global and local exemplars, assigning a forgetting ability to particles and adjusting the acceleration coefficients of particles based on the population's experience in the swap sequence based PSO in order to improve the performance of the swap sequence based PSO.

From the results obtained, the new algorithm is shown to provide better solutions than the swap sequence based PSO. However, in terms of convergence rate, the swap sequence based PSO is better since it takes a shorter time to converge to the optimal solution than the new algorithm.

For future work, the technique of multidimensional PSO can be included in the algorithm so that other parameters can be taken into consideration. In this paper, the most efficient route, which can be taken by a driver while delivering items from a shipping company, is obtained by finding the shortest route in terms of the distance. With multidimensional PSO, other factors such as the traffic level, road condition, and traffic lights can also be taken into account. Moreover, a tool with a user interface can be proposed which implements this algorithm and allows users to enter the locations by themselves. After taking in the locations from the user via the user interface, the algorithm is run to find the shortest path which is then displayed to the user via the user interface.

REFERENCES

[1] G. Gutin and A. Punnen, *The Traveling Salesman Problem and its Variations*. Boston, MA, USA: Springer, 2007.

[2] D. Davendra, *Traveling Salesman Problem, Theory and Applications*. Rijeka, Croatia: IntechOpen, 2010.

[3] C. Rego, D. Gamboa, F. Glover, and C. Osterman, "Traveling salesman problem heuristics: Leading methods, implementations and latest advances," *Eur. J. Oper. Res.*, vol. 211, no. 3, pp. 427–441, Jun. 2011.

[4] S. Almufti, "Single-based and population-based metaheuristics algorithms performances in solving NP-hard problems," *Iraqi J. Sci.*, vol. 62, pp. 1710–1720, 2021.

[5] C. Blum and X. Li, "Swarm intelligence in optimization," in *Swarm Intelligence* (Natural Computing Series). Berlin, Germany: Springer, 2008.

[6] A. Chakraborty and A. K. Kar, "Swarm intelligence: A review of algorithms," in *Nature-Inspired Computing and Optimization* (Modeling and Optimization in Science and Technologies). Cham, Switzerland: Springer, 2017.

[7] E. Osaba, *Applied Optimization and Swarm Intelligence* (Springer Tracts in Nature-Inspired Computing). Singapore: Springer, 2017. [Online]. Available: https://books.google.com.my/books?id=1xkvEAAAQBAJ

[8] W. Elloumi, H. El Abed, A. Abraham, and A. M. Alimi, "A comparative study of the improvement of performance using a PSO modified by ACO applied to TSP," *Appl. Soft Comput.*, vol. 25, pp. 234–241, Dec. 2014.

[9] R. Poli, J. Kennedy, and T. Blackwell, "Particle swarm optimization," *Swarm Intell.*, vol. 1, no. 1, pp. 33–57, Jun. 2007.

[10] M. A. El-Shorbagy and A. E. Hassanien, "Particle swarm optimization from theory to applications," *Int. J. Rough Sets Data Anal.*, vol. 5, no. 2, pp. 1–24, 2018.

[11] S. S. Rao and P. Siddaiah, "Design of eight-phase sequences using modified particle swarm optimization for spread spectrum and radar applications," *J. Comput. Sci. Inf. Technol.*, vol. 6, p. 30, May 2021.

[12] N. Fatema and H. Malik, "Data-driven occupancy detection hybrid model using particle swarm optimization based artificial neural network," in *Metaheuristic and Evolutionary Computation: Algorithms and Applications* (Studies in Computational Intelligence). Singapore: Springer, 2020.

[13] J. Zeng and C. Wang, "A novel hyperchaotic image encryption system based on particle swarm optimization algorithm and cellular automata," *Secur. Commun. Netw.*, vol. 2021, pp. 1–15, Feb. 2021.

[14] A. Mullai and K. Mani, "Enhancing the security in RSA and elliptic curve cryptography based on addition chain using simplified swarm optimization and particle swarm optimization for mobile devices," *Int. J. Inf. Technol.*, vol. 13, no. 2, pp. 551–564, Apr. 2021.

[15] X. Xia, L. Gui, G. He, B. Wei, Y. Zhang, F. Yu, H. Wu, and Z.-H. Zhan, "An expanded particle swarm optimization based on multi-exemplar and forgetting ability," *Inf. Sci.*, vol. 508, pp. 105–120, Jan. 2020.

[16] K.-P. Wang, L. Huang, C.-G. Zhou, and W. Pang, "Particle swarm optimization for traveling salesman problem," in *Proc. Int. Conf. Mach. Learn. Cybern.*, 2003, pp. 1583–1585.

[17] A. Mah, S. I. Hossain, and S. Akter, "A comparative study of prominent particle swarm optimization based methods to solve traveling salesman problem," *Int. J. Swarm Intell. Evol. Comput.*, vol. 5, no. 3, p. 139, 2016.

[18] M. A. H. Akhand, S. Akter, S. S. Rahman, and M. M. H. Rahman, "Particle swarm optimization with partial search to solve traveling salesman problem," in *Proc. Int. Conf. Comput. Commun. Eng. (ICCCE)*, Jul. 2012, pp. 118–121.

[19] J.-W. Zhang and W.-J. Si, "Improved enhanced self-tentative PSO algorithm for TSP," in *Proc. 6th Int. Conf. Natural Comput.*, Aug. 2010, pp. 2638–2641.

[20] I. Khan, M. K. Maiti, and M. Maiti, "Coordinating particle swarm optimization, ant colony optimization and K-opt algorithm for traveling salesman problem," in *Communications in Computer and Information Science* (Mathematics and Computing). Singapore: Springe, 2017.

[21] B. Cheng, H. Lu, Y. Huang, and K. Xu, "An improved particle swarm optimization algorithm based on Cauchy operator and 3-opt for TSP," in *Proc. 17th Int. Conf. Parallel Distrib. Comput., Appl. Technol. (PDCAT)*, Dec. 2016, pp. 177–182.

[22] Y. Zhong, J. Lin, L. Wang, and H. Zhang, "Discrete comprehensive learning particle swarm optimization algorithm with metropolis acceptance criterion for traveling salesman problem," *Swarm Evol. Comput.*, vol. 42, pp. 77–88, Oct. 2018.

[23] Z. Hua, J. Chen, and Y. Xie, "Brain storm optimization with discrete particle swarm optimization for TSP," in *Proc. 12th Int. Conf. Comput. Intell. Secur. (CIS)*, Dec. 2016, pp. 190–193.

[24] T. Lai, H. Sun, and H. Li, "A novel discrete particle swarm optimization for travelling salesman problem based on dissipative structure theory," in *Proc. 3rd Int. Conf. Unmanned Syst. (ICUS)*, Nov. 2020, pp. 423–427.

[25] Y. Wang, "Improving artificial bee colony and particle swarm optimization to solve TSP problem," in *Proc. Int. Conf. Virtual Reality Intell. Syst. (ICVRIS)*, Aug. 2018, pp. 179–182.

[26] J. Xin, S. Li, Y. Zhang, Y. Cui, and J. Sheng, "Application of improved particle swarm optimization for navigation of unmanned surface vehicles," *Sensors*, vol. 19, no. 14, p. 3096, 2019.

[27] Q. Lv and D. Yang, "Multi-target path planning for mobile robot based on improved PSO algorithm," in *Proc. IEEE 5th Inf. Technol. Mechatronics Eng. Conf. (ITOEC)*, Jun. 2020, pp. 1042–1047.

[28] B. Wei, Y. Xing, X. Xia, and L. Ling, "A novel particle swarm optimization with genetic operator and its application to tsp," *Int. J. Cogn. Inform. Natural Intell.*, vol. 15, no. 4, pp. 1–17, 2021.

[29] Y. Cui, J. Zhong, F. Yang, S. Li, and P. Li, "Multi-subdomain grouping-based particle swarm optimization for the traveling salesman problem," *IEEE Access*, vol. 8, pp. 227497–227510, 2020.

[30] A. M. Alimi, I. Twir, N. Rokbani, and P. Kromer, "A new hybrid gravitational particle swarm optimisation-ACO with local search mechanism, PSOGSA-ACO-Ls for TSP," *Int. J. Intell. Eng. Informat.*, vol. 7, no. 4, p. 384, 2019.

BIBI AAMIRAH SHAFAA EMAMBOCUS received the B.Sc. degree in computer science from Sunway University, Malaysia, in 2020, where she is currently pursuing the M.Sc. degree in computer science. Her research interests include swarm intelligence, evolutionary algorithms, artificial intelligence, and machine learning.

MUHAMMED BASHEER JASSER (Member, IEEE) received the master's and Ph.D. degrees in software engineering from University Putra Malaysia (UPM). He is currently a Program Leader and a Lecturer at Sunway University. Prior to that, he was a Research Assistant at UPM and a Lecturer at Aleppo University. He was granted the Malaysian Technical Cooperation Program (MTCP) Scholarship from the Ministry of Higher Education, Malaysia, for his postgraduate studies. He is also working on several fundamental and industrial research projects in the area of artificial intelligence and software engineering funded by companies and universities. His major research interests include optimization algorithms, evolutionary computation, model-driven software engineering, formal specification, verification and theorem proving, artificial intelligence, and machine learning. He is a member of several professional academic bodies, including IEEE, the Institute of Electronics, Information and Communication Engineers (IEICE), and Formal Methods Europe.

MUZAFFAR HAMZAH received the B.Sc. and M.Sc. degrees in computer science from the University of Malaya and the Ph.D. degree in information technology from the University of South Australia. He has also completed his Postdoctoral Research in geovisualization with Nottingham University. He is currently the Deputy Dean (Graduate and Internationalization) and a Senior Lecturer at the Faculty of Computing and Informatics, Universiti Malaysia Sabah, Malaysia. He actively involved in research about HCI and information visualization. His current research interests include information theory, visual analytics, and formal methods. He has also led several research grants at national level as a principal investigator and recently recognized by IBIMA for the Best Paper Award.

AIDA MUSTAPHA (Member, IEEE) received the B.Sc. degree in computer science from Michigan Technological University and the Ph.D. degree in artificial intelligence from Universiti Putra Malaysia. Her research interests include soft computing, data mining, computational linguistics, and software agents. Her main passion is pursuing the area of natural language studies and application of natural language processing to holy texts, in particular the Quran.

ANGELA AMPHAWAN (Member, IEEE) received the Ph.D. degree in optical engineering from the University of Oxford, U.K. She is currently leading the Photonics Research Laboratory at the School of Engineering and Technology, Sunway University. Prior to this, she was the Deputy Vice Chancellor of the University Malaysia of Computer Science and Engineering. Her research projects have been funded by the U.S. Department of States, German Government, Malaysian Ministry of Education, and Telekom Malaysia. Her research interests include optical communications and sensing, covering optical fibers, free-space optics, radio-over free space optics, digital imaging, and the Internet of Things. She has won several Best Paper Awards and exhibition medals. She was a recipient of the Fulbright Award from the Research Laboratory of Electronics, Massachusetts Institute of Technology. She serves on the Editorial Board of the *APL Photonics* journal with the American Institute of Physics. She is on the National 5G Task Force. She was previously on the Editorial Board of *Transactions on Emerging Telecommunications Technologies* (Wiley) and the Publicity Co-Chair of the IEEE Wireless Communications and Networking Conference. She has given keynote addresses at several Fulbright and IEEE events.