

# A NEW HYBRID YIN-YANG-PAIR-PARTICLE SWARM OPTIMIZATION ALGORITHM FOR UNCAPACITATED WAREHOUSE LOCATION PROBLEMS

A. A. Heidari\*, O. Kazemizade, F. Hakimpour

School of Surveying and Geospatial Engineering, College of Engineering, University of Tehran, Iran  
(as\_heidari, kazemizade.omid, fhakimpour)@ut.ac.ir

**KEYWORDS:** Warehouse Location; Optimal; Location Analysis; Optimization; Particle Swarm; Yin-yang-pair Optimization

## ABSTRACT:

Yin-Yang-pair optimization (YYPO) is one of the latest metaheuristic algorithms (MA) proposed in 2015 that tries to inspire the philosophy of balance between conflicting concepts. Particle swarm optimizer (PSO) is one of the first population-based MA inspired by social behaviors of birds. In spite of PSO, the YYPO is not a nature inspired optimizer. It has a low complexity and starts with only two initial positions and can produce more points with regard to the dimension of target problem. Due to unique advantages of these methodologies and to mitigate the immature convergence and local optima (LO) stagnation problems in PSO, in this work, a continuous hybrid strategy based on the behaviors of PSO and YYPO is proposed to attain the suboptimal solutions of uncapacitated warehouse location (UWL) problems. This efficient hierarchical PSO-based optimizer (PSOYPO) can improve the effectiveness of PSO on spatial optimization tasks such as the family of UWL problems. The performance of the proposed PSOYPO is verified according to some UWL benchmark cases. These test cases have been used in several works to evaluate the efficacy of different MA. Then, the PSOYPO is compared to the standard PSO, genetic algorithm (GA), harmony search (HS), modified HS (OBCHS), and evolutionary simulated annealing (ESA). The experimental results demonstrate that the PSOYPO can reveal a better or competitive efficacy compared to the PSO and other MA.

## 1. INTRODUCTION

The suitability of civic warehouses is influenced by their locations. One of the vital concerns in supply chain management, logistic and location analysis is warehouse location (FL) (Ou-Yang and Ansari, 2017). In recent years, FL problems, which is also termed in the related works as facility location (FL), have been extensively investigated by researchers because of their tactical nature (Aardal et al., 1999; Ou-Yang and Ansari, 2017). There are various mathematical models demonstrating a range of FL tasks, which most of them have a combinatorial nature (Aardal et al., 1999). Consequently, exact algorithms can often be effective for small instances and metaheuristic algorithms (MA) have been utilized in literature as efficient methods to handle larger applied FL tasks.

In the traditional uncapacitated WL (UWL), each warehouse requires an initial cost to be established and a certain cost is associated with the choosing of roads from a client to a warehouse. The objective of the UWL is to determine where to construct the warehouses and which routes to utilize for optimizing the overall costs. In UWL, the decision maker should determine the candidate sites that specify a new warehouse, while some criteria such as total costs to be minimized (Al-Sultan and Al-Fawzan, 1999; Korupolu et al., 2000). The cost includes fixed costs to open plants and depots, and variable cost can be related to the transportation. The other constraints can be the demands of all stores and the warehouse and suppliers capacity limits, which should not be violated. Therefore, the selection of warehouse location will profoundly influence the management planning of plants and organizations. It's worth noting that several applied tasks with no warehouses to obtain, such as portfolio management, machine scheduling, clustering, and computer networks can similarly be addressed according to UWL-based models (Ghosh, 2003; Tcha and Lee, 1984).

Some exact approaches have been developed for solving the UWL. Some of the well-known methods are branch and bound (Klose, 1998), dual approach (DUALLOC) (Erlenkotter, 1978) Lagrangean relaxation (Barcelo et al., 1990), the primal-dual (Körkel, 1989), and linear programming. It can be proved that the UWL is NP-hard (Sevкли and Guner, 2006). Hence, in last 10 years, many researchers tried to develop efficient MA to tackle the UWL. Some of the well-established works can be summarized as tabu search (TS) (Michel and Van Hentenryck, 2004), simulated annealing (SA) (Aydin and Fogarty, 2004), particle swarm optimizer (PSO) (Guner and Sevкли, 2008), and genetic algorithms (GA) (Jaramillo et al., 2002). In 2017, Guo et al proposed a two-stage capacitated FL (TSCFLP) and a hybrid evolutionary algorithm (EA) (Guo et al., 2017). In 2015, Basti and Sevкли used artificial bee colony (ABC) optimizer to realize the p-median UFL problem (Basti and Sevкли, 2015). Moreover, Heidari et al proposed an efficient opposition-based chaotic HS (OBCHS) to tackle the UFL problems and the results revealed that the OBCHS can reveal a better efficacy compared to GA and PSO methods (Heidari et al., 2015a). Ng investigated expanding neighborhood tabu search (ENTS) algorithm for large warehouse location problems in water infrastructure planning (Ng, 2014). Esnaf et al. proposed a fuzzy c-means algorithm with fixed cluster centers for UFL that permits unlabeled data to be assigned to the related clusters centers. In their research, the proposed algorithm is tested on different UFL benchmarks from literature and Turkish fertilizer producer's real data. The algorithm was compared to the PSO and ABC algorithms and the results show that the proposed algorithm have better performance than the other algorithms (Esnaf et al., 2014). In 2017, a hybrid PSO-tabu search optimizer has been proposed to treat the hierarchical FL problem (Ou-Yang and Ansari, 2017). The Yin-Yang-pair optimization (YYPO) is one of the recent MA that mimics the philosophy of balance among discordant concepts (Punnathanam and Kotecha, 2017).

\* Corresponding author

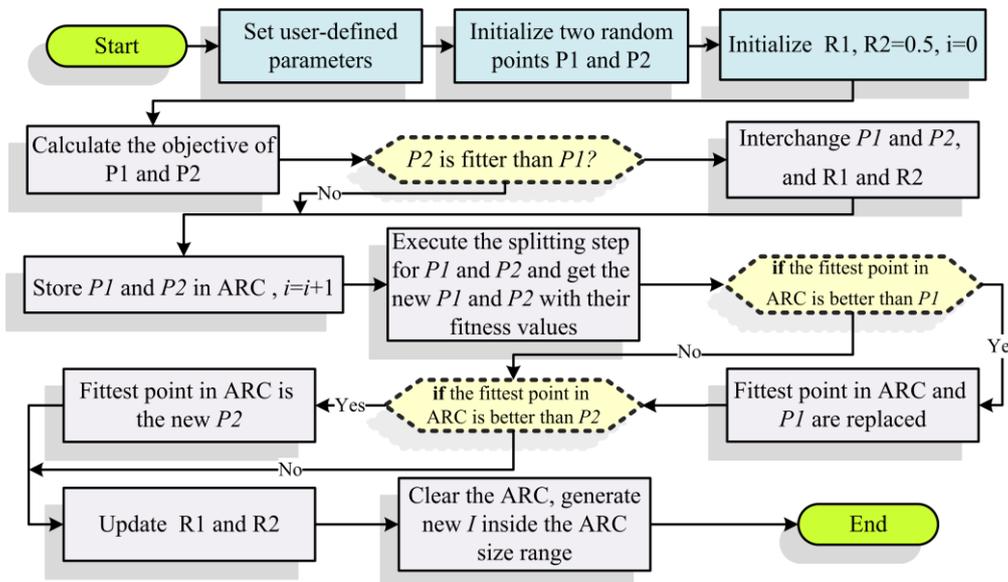


Figure 1. The structure of YYPO algorithm

This algorithm has shown an efficient performance in dealing with several optimization problems. In this paper, a hybrid MA is proposed and substantiated to tackle the UWL problem. For this purpose, the YYPO optimizer is embedded into the well-known PSO algorithm to enhance the efficacy of this algorithm. Then, the proposed PSO-YYPO is validated based on several instances of the UWL.

## 2. THE YYPO ALGORITHM

The YYPO optimizer is one of the recent MA inspired by the philosophy of balance among discordant concepts (Punnathanam and Kotecha, 2017). In YYPO, the variables of the tasks should be normalized inside interval (0, 1). This optimizer utilizes two points (P1 and P2) to search the problem landscape. The P1 and P2 are generated in the initial step of YYPO inside the domain of [0, 1] and their fitness are evaluated. The better point is considered as P1 and the other one is assigned as P2. The point P1 plays its role in exploitation phase, while the point P2 tries to highlight the exploration behaviors. The points P1 and P2 act as hubs to sightsee the hypersphere dimensions inside the solution space specified by radii of R1 and R2, correspondingly. These spans have a self-adaptive nature such that R1 has a propensity to every so often decrease and R2 to rise. The YYPO involves two core phases:

- Splitting phase
- Archive phase

### 2.1 Splitting phase

This phase is designed to boost the exploration of the target space. For this purpose, one of the points in consort with its search range is inserted to this step. The splitting phase can create new positions inside the hypersphere at diverse directions. In this step, 2D fresh points are scattered nearby the particular point P and their distances to the P is calculated based on R. Note that D denotes the dimension. The new points can be scattered using two strategies with the same chance: one-way and D-way splitting. In one-way strategy, simply one variable of each P is adjusted. In the D-way method, all variables of each P are updated (Punnathanam and Kotecha, 2017).

One-way splitting: 2D duplicates of the P are kept as S then; one variable of each P in S is adjusted by Eq. (1):

$$S_j^i = S_j^i + a \times R \quad (1)$$

$$S_{D+j}^i = S_j^i - a \times R, j = 1, 2, 3, \dots, D \quad (2)$$

In aforementioned relations, superscript shows the variable number, the subscript are the P number and *a* shows a random value inside (0, 1). In D-way splitting step, all variables of P in S should be updated using:

$$\begin{cases} S_k^j = S_j^i + a(R/2^{0.5}) & \text{if } B_k^j = 1 \\ S_k^j = S_j^i + a(R/2^{0.5}) & \text{else} \end{cases} \quad (3)$$

$k = 1, 2, 3, \dots, 2D. \quad j = 1, 2, 3, \dots, D$

Here, the binary matrix *B* is made by randomly choosing 2D distinctive integers between 0 and  $2^{D-1}$ .

### 2.2 Archive stage

Initialization of this step indicates that *I* archive (ARC) updates have been completed. Moreover, the archive comprises 2*I* points. The finest point in the ARC, when it is superior to *P1*, is switched with *P1* (i.e. the former *P1* is currently in the ARC). Then, the best point in the present ARC, if superior to *P2*, substitutes *P2*. Thus, the best two points will continuously survive sequential repetitions (Punnathanam and Kotecha, 2017). The R1 and R2 are updated as:

$$R1 = R1 - (R1/\alpha) \quad (4)$$

$$R2 = R2 + (R2/\alpha) \quad (5)$$

The *R1* can shrinkage the volume explored nearby *P1* in the splitting phase, while *R2* can expand the volume explored nearby *P2*. In the YYPO, the maximum of *R2* is 0.75. Then, the ARC is deflated and a renewed value of *I* is created inside ( $I_{min}$  and  $I_{max}$ ).

The flowchart of YYPO is demonstrated in Figure 1. For more details about the YYPO please refer to (Punnathanam and Kotecha, 2017).

### 3. THE PSO ALGORITHM

The PSO is a well-known population-based MA used in numerous works (Heidari et al., 2017a, b; Heidari and Delavar, 2016; Heidari et al., 2015b; Trelea, 2003). The PSO tries to simulate the idealistic social life of birds (Heidari and Pahlavani, 2017c). The PSO utilizes a swarm of particles (solutions) that can search the target domain to explore and exploit the fittest particle. Temporarily, the search agents all follow the best solution in their searching trajectories. In PSO, particles will track their personal best locations along with the best agent explored thus far. The PSO can be mathematically described as:

$$v_i^{t+1} = \omega v_i^t + c_1 \varphi_1 (p_i - x_i^t) + c_2 \varphi_2 (p_g - x_i^t) \quad (6)$$

$$x_i^{t+1} = x_i^t + v_i^{t+1} \quad (7)$$

where  $v_i^t$  denotes the velocity of agent  $i$  at iteration  $t$ ,  $w$  shows a weighting factor,  $c_j$  is an acceleration factor,  $\varphi_1$  and  $\varphi_2$  are random values inside (0, 1),  $x_i^t$  is the existing location of agent  $i$  at iteration  $t$ ,  $p_i$  is the  $pbest$  of solution  $i$  at loop  $t$ , and  $p_g$  is the fittest agent so far. A modified PSO (CFM) is also available that utilizes a constriction factor (CF) (Heidari and Ali Abaspour, 2016). In this paper, the CFM version is employed to treat the UWL problem. The main equation of CFM can be described as:

$$v_i(t) = \chi [v_i(t) + c_1 \varphi_1 (p_i - x_i(t)) + c_2 \varphi_2 (p_g - x_i(t))] \quad (8)$$

$$x_i(t+1) = x_i(t) + v_i(t) \quad (9)$$

$$\chi = 2 / |2 - \varphi - (\varphi^2 - 4\varphi)^{0.5}|, \quad \varphi = c_1 + c_2, \quad \varphi > 4, \quad (10)$$

where  $\chi$  denotes the constriction factor.

### 4. THE HIERARCHICAL PSO-YYPO ALGORITHM

The basic PSO can reveal a satisfactory performance in treating different optimization tasks. However, immature convergence to LO may still happen in tackling some larger UWL cases. The reason is that PSO cannot make a stable balance between exploration and exploitation tendencies. In order to mitigate this drawback, an enhanced PSO algorithm is proposed to enrich its efficacy on UWL tasks. For this purpose, the YYPO optimizer can be used an extra step inside the PSO method.

The main structure of combined algorithm (PSOYPO) is similar to the basic PSO. The first difference is that the global best solution of PSO is obtained during iterations via YYPO and inserted into the PSO again for more improvements. The second difference is the hierarchical structure of PSOYPO. In the initial step of PSO, the YYPO is started to generate the best results. Then, the outputs of YYPO are considered as the initial candidate solutions for PSO. The third difference is that the PSO will search the topography of the target problem, then, the YYPO is triggered again improve the local searching capacity of PSO. After that, the basic PSO can continue the search until the last iteration. In PSOYPO, the operators of YYPO have a constructive effect on the precision of solutions based on the two best agents in sorted population. It's worth noting that YYPO is not a population-based optimizer and improves only

two solutions during the searching process. The YYPO is a low complexity MA which starts with two initial positions and produces more points with regard to the dimension of target problem. Hence, when the YYPO wants to be utilized, the population of particles is sorted and only the first and second best agents are inserted as the inputs to the YYPO.

The proposed PSOYPO has the main advantages of both PSO and YYPO. In this work, PSOYPO is used to handle the UWL problem. To learn how the PSO, PSOYPO, and other compared methods are applied to UWL problems, the reader is referred to (Guner and Sevkli, 2008; Sevkli and Guner, 2006).

### 5. UWL PROBLEM

In the UWL, a subset of locations from a given set of potential locations is required for establishing warehouses so as to optimize a given function of these chosen locations while satisfying certain constraints (Basu et al., 2015; Michel and Van Hentenryck, 2004). In this section, the variables of the UWL problem are described.

In UWL problems with  $m$  stores and  $n$  candidate warehouse sites,  $f_j$  is used to represent the cost of opening warehouse  $j$  that is fixed and  $c_{ij}$  is used to represent the cost of serving customer  $i$  from warehouse  $j$  or assigning store  $i$  to warehouse  $j$ . We assume that  $c_{ij} \geq 0$  for all  $i = 1, 2, 3 \dots m$  and  $j = 1 \dots n$  and  $f_j > 0$  for all  $j = 1 \dots n$ . A binary variable  $y_j$  is used to represent the status of warehouse  $j$  in the model. Warehouse  $j$  will be opened only if  $y_j = 1$  in the solution. A binary variable  $x_{ij}$  is used for the road from store  $i$  to warehouse  $j$  in the model. Store  $i$  will be served by warehouse  $j$  only if  $x_{ij} = 1$  in the solution. However, each  $x_{ij}$  can be treated as a continuous variable and will have a binary value in the solution. The solution process of the UWL problem is to find an optimal solution that satisfies the store demand and minimizes the total cost (Esnaf et al., 2014; Sun, 2006). Here, the formally UWL is described as:

$$\text{Minimize } Z = \sum_{i=1}^n f_{ci} \cdot y_i + \sum_{i=1}^m \sum_{j=1}^n c_{ij} \cdot x_{ij} \quad (11)$$

$$\text{s.t. } \sum_{i=1}^n x_{ij} = 1, \quad (12)$$

$$0 \leq x_{ij} \leq y_i, \quad y_i \in \{0,1\}, \quad (13)$$

where  $i = 1, \dots, n; j = 1, \dots, m;$

$x_{ij}$ : the quantity provided using facility  $i$  to client  $j$ ;

$y_i$ : whether facility  $i$  is confirmed ( $y_i = 1$ ) or not ( $y_i = 0$ )

The UWL problems usually have a mixed integer formulation with binary variables to indicate the locations of warehouses, together with continuous variables that represent system dynamics. This mixed integer nature of UWL problems makes them NP-hard (Ng, 2014; Sevkli and Guner, 2006). The solution methodologies used for these problems concentrate on exact methods and MA. Because the UWL is NP-hard, exact methods are mostly suitable for small-sized UWL cases. In the large-scale cases, MA are used in order to obtain near optimum solutions since finding optimum solutions need much time (Basti and Sevkli, 2015).

In addition, several MA have been applied for tackling UWL problems such as hybrid EA (Guo et al., 2017), ABC (Basti and Sevkli, 2015), an opposition-based chaotic harmony search algorithm (Heidari et al., 2015a), tabu search (Sun, 2006),

expanding neighborhood tabu search (Ng, 2014), a simplified artificial fish swarm (Azad et al., 2013), and a modified continuous PSO (Saha et al., 2011). In the next sections, the results of a new effective PSO-based optimizer are reported to investigate the optimum solutions of several UWL problems.

## 6. RESULTS AND ANALYSIS

In this section, the efficacy of the new PSOYPO is studied carefully. Here, each method is experienced using MATLAB 2013 software. The PSOYPO is utilized to solve 15 well-known UWL benchmark test cases from OR Library (Beasley, 2005). Twelve test problems are somewhat small in size, while three test cases are pretty large. The details of tackled benchmark problems are reported in Table 1.

ID	Name	Type	Size ( $m \times n$ )	Optimum
TF01	Cap71	Small	16 × 50	932615.8
TF02	Cap72	Small	16 × 50	977799.4
TF03	Cap73	Small	16 × 50	1010641
TF04	Cap74	Small	16 × 50	1034977
TF05	Cap101	Small	25 × 50	796648.4
TF06	Cap102	Small	25 × 50	854704.2
TF07	Cap103	Small	25 × 50	893782.1
TF08	Cap104	Small	25 × 50	928941.8
TF09	Cap131	Small	50 × 50	793439.6
TF10	Cap132	Small	50 × 50	851495.3
TF11	Cap133	Small	50 × 50	893076.7
TF12	Cap134	Small	50 × 50	928941.8
TF13	Cap-A	Large	100 × 1000	17156454
TF14	Cap-B	Large	100 × 1000	12979072
TF15	Cap-C	Large	100 × 1000	11505594

Table 1. Details of benchmark test cases

Each method is verified based on 30 independent trials. The population size of all methods is equal to the number of warehouses. Note that the recommended settings in related literature are used in this experiment (Heidari et al., 2015a; Sevкли and Guner, 2006). In addition, the algorithms can repeat the process throughout  $1.00E+03$  iterations. In this research, the mean relative percent error (*ARPE*) is measured as the performance index. The *MRPE* is described as:

$$MRPE = \sum_{i=1}^Z \left( \frac{(T_i - S) \times 100}{S_i} \right) / Z \quad (14)$$

where  $T_i$  is the  $i$ th iteration,  $U$  denotes the fittest value and  $Z$  represents the iteration number. The other indexes utilized in this study are "Optimum Rate" represented by (*HR*) and CPU time indicated by (*RT*). Furthermore, *HR* is the hit to the optimum rate (*HR*) ratio that shows the robustness of algorithms. It is truly hard to explore the best positions in every round of search. The higher *HR* and *MRPE* value are better than the lower values. When these indexes are higher, it can be recognized that the quality of results is more satisfactory.

The *MRPE* values of the PSOYPO algorithm are compared to HS, OBCHS, and PSO in Table 2. It can be seen from Table 2 that the proposed PSOYPO can obtain preferable solutions compared to the basic PSO method. It is seen that for problems with bigger sizes, the error rate of all methods has increased. The reason that PSOYPO can obtain results with a better precision is that it has an effective exploitation capacity. Hence, in the case of finding a fruitful region, it can focus on the better positions in the search space. The *MPRE* values of PSOYPO are very competitive to the OBCHS, but OBCHS still can show

a better efficacy than PSO-based methods for TS13, TS14, and TS15 problems.

ID	PSO	HS	OBCHS	PSOYPO
TS1	6.00E-02	4.00E-02	0.00E+00	0.00E+00
TS2	8.00E-02	9.00E-02	0.00E+00	0.00E+00
TS3	7.00E-02	7.00E-02	1.00E-02	0.00E+00
TS4	8.00E-02	8.00E-02	0.00E+00	0.00E+00
TS5	1.60E-01	1.30E-01	0.00E+00	2.00E-02
TS6	1.70E-01	1.00E-01	0.00E+00	0.00E+00
TS7	1.50E-01	1.70E-01	2.00E-02	3.00E-02
TS8	1.70E-01	1.90E-01	0.00E+00	0.00E+00
TS9	7.90E-01	7.60E-01	0.00E+00	0.00E+00
TS10	8.10E-01	7.70E-01	1.00E-02	8.00E-02
TS11	7.10E-01	7.20E-01	0.00E+00	0.00E+00
TS12	9.10E-01	9.40E-01	0.00E+00	2.00E-02
TS13	2.12E+01	1.27E+01	1.00E-02	1.00E-02
TS14	9.91E+00	4.12E+00	1.00E-02	2.41E+00
TS15	9.62E+00	2.41E+00	3.00E-02	4.52E+00

Table 2. Comparison of *MRPE* values

In Table 3, The *HR* values of the PSOYPO algorithm are compared to the HS, OBCHS, and PSO algorithms (Also see Figure 2).

ID	PSO	HS	OBCHS	PSOYPO
TS1	8.50E-01	9.00E-01	1.00E+00	1.00E+00
TS2	7.90E-01	8.40E-01	1.00E+00	1.00E+00
TS3	6.50E-01	6.80E-01	1.00E+00	1.00E+00
TS4	7.10E-01	7.80E-01	1.00E+00	1.00E+00
TS5	5.60E-01	4.20E-01	1.00E+00	1.00E+00
TS6	4.60E-01	5.40E-01	1.00E+00	9.80E-01
TS7	1.90E-01	1.20E-01	1.00E+00	1.00E+00
TS8	6.50E-01	7.20E-01	9.90E-01	9.90E-01
TS9	6.00E-02	7.00E-02	1.00E+00	1.00E+00
TS10	0.00E+00	1.00E-02	1.00E+00	9.80E-01
TS11	0.00E+00	0.00E+00	1.00E+00	1.00E+00
TS12	1.00E-01	1.40E-01	1.00E+00	1.00E+00
TS13	0.00E+00	0.00E+00	9.80E-01	1.00E+00
TS14	0.00E+00	0.00E+00	9.70E-01	9.70E-01
TS15	0.00E+00	0.00E+00	5.00E-01	5.60E-01

Table 3. Comparison of *HR* values

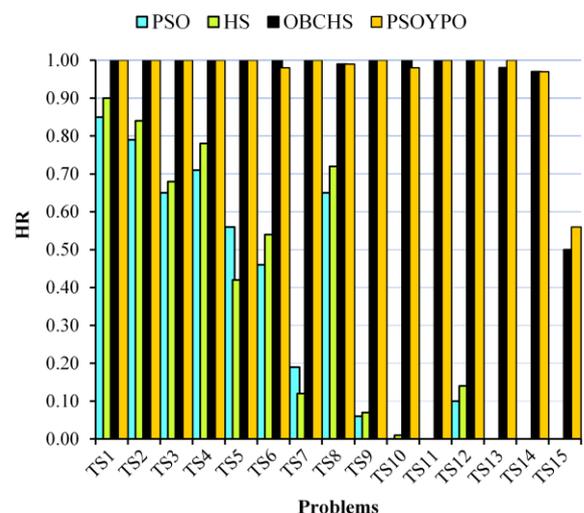


Figure 2. Robustness comparison between the PSO, HS, OBCHS, and PSOYPO algorithms

It can be seen that PSOYPO can obtain better results than PSO in dealing with all problems. The results of the PSOYPO method are very competitive to the OBCHS, but it is observed that the results are better than basic HS. Generally, the PSO cannot show an outstanding effectiveness in comparison with HS, OBCHS, and PSOYPO.

In Table 3, the *RT* values of the PSOYPO algorithm are compared to HS, OBCHS, and PSO. It is seen that the PSOYPO require additional time obtain better results compared to the PSO (see Figures 3 and 4). The reason is that the proposed strategy has a hierarchical structure that increases the time of the process. However, OBCHS is still the fastest method.

ID	PSO	HS	OBCHS	PSOYPO
TS1	1.30E-01	1.20E-01	7.00E-02	1.50E-01
TS2	1.50E-01	1.90E-01	6.00E-02	2.10E-01
TS3	2.80E-01	2.30E-01	1.40E-01	3.10E-01
TS4	2.30E-01	1.05E+00	4.50E-01	2.60E-01
TS5	7.10E-01	6.50E-01	6.10E-01	8.50E-01
TS6	8.00E-01	9.80E-01	8.50E-01	9.60E-01
TS7	1.13E+00	9.90E-01	7.80E-01	1.23E+00
TS8	5.10E-01	4.50E-01	1.20E-01	6.50E-01
TS9	4.34E+00	4.13E+00	2.41E+00	5.23E+00
TS10	4.62E+00	4.14E+00	3.98E+00	6.62E+00
TS11	4.51E+00	4.13E+00	3.41E+00	5.52E+00
TS12	4.53E+00	3.98E+00	2.12E+00	4.87E+00
TS13	1.47E+01	1.47E+01	8.12E+00	1.56E+01
TS14	1.77E+01	1.60E+01	1.30E+01	1.84E+01
TS15	2.57E+01	2.24E+01	1.90E+01	2.93E+01

Table 4. Comparison of *RT* values

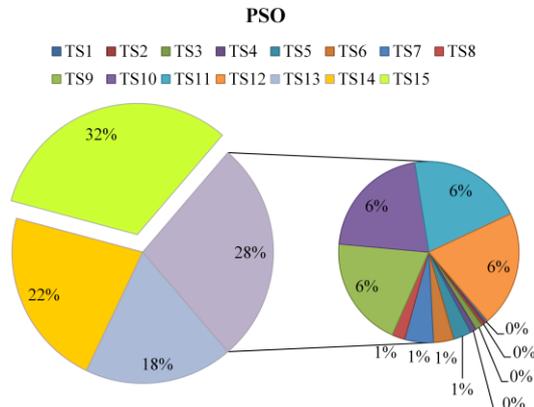


Figure 3. Computational efficacy of the PSO in dealing with TS1 to TS15

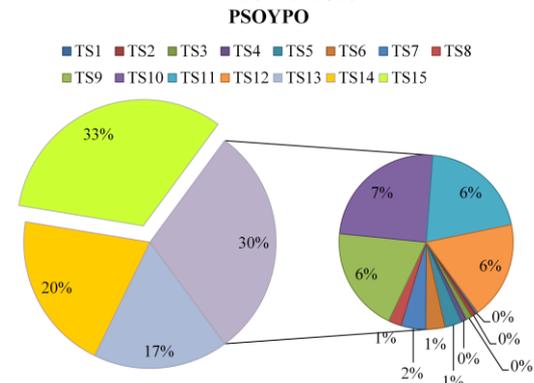


Figure 4. Computational efficacy of the proposed PSOYPO in dealing with TS1 to TS15

Regarding the measured indexes, it is seen that the efficiency of PSOYPO is superior to PSO and other methods. The reason is that it has an enhanced local search potential, which can help it in escaping from LO. To further analysis of the performance of the PSOYPO, it is compared with a GA-based strategy proposed by (Jaramillo et al., 2002) and evolutionary SA (ESA) developed by (Aydin and Fogarty, 2004). Note that the results of GA and ESA are obtained and reported here based on their original works (Sevкли and Guner, 2006). The standard deviation (STD) results of different algorithms for normal scale (NS) and large scales (LS) problems are compared in Tables 5 and Table 6, respectively. In addition, the running time of different methods is tabulated in Table 7 and Table 8. Note that the HS, OBCHS, PSO, and PSOYPO algorithms were tested on the same computer.

ID	GA	ESA	HS	OBCHS	PSO	PSOYPO
TS1	0.00	0.00	0.00	0.00	0.00	0.00
TS2	0.00	0.00	0.001	0.00	0.001	0.00
TS3	0.00033	0.00	0.00021	0.0001	0.002	0.00
TS4	0.00	0.00	0.00	0.00	0.002	0.002
TS5	0.00020	0.00	0.0001	0.00	0.005	0.003
TS6	0.00	0.00	0.0001	0.00	0.001	0.00
TS7	0.00015	0.00	0.0023	0.00	0.052	0.00
TS8	0.00	0.00	0.00	0.00	0.014	0.01
TS9	0.00065	0.00008	0.0004	0.00	0.004	0.00
TS10	0.00	0.00	0.00	0.00	0.005	0.00
TS11	0.00037	0.00002	0.00075	0.00	0.002	0.00
TS12	0.00	0.00	0.00	0.00	0.001	0.00

Table 5. Comparison of STD results for normal scale problems

ID	GA	ESA	HS	OBCHS	PSO	PSOYPO
TS13	0.00	0.000	0.00	0.00	0.001	0.00
TS14	0.00172	0.00070	0.001	0.00002	0.001	0.00001
TS15	0.00131	0.00119	0.00	0.00	0.051	0.00

Table 6. Comparison of STD results for LS problems

From the results in Table 5, it is noticeable that the results of PSOYPO algorithm have a superior accuracy. The STD results are lower than other methods in the majority of test cases. It is seen that for all problems, except TS4, TS5, and TS8, the proposed PSOYPO can show an impressive performance according to the STD results and its results have the least deviation from the optimum. It's also seen that the accuracy of results of the PSO isn't impressive. From Table 6, it is seen that the HS, OBCHS, and PSOYPO algorithms can obtain the best results in solving TS14 test case (see Figure 5). The overall outcomes confirm that the stagnation problems of PSO have been alleviated significantly based on the proposed YYPO-based mechanisms.

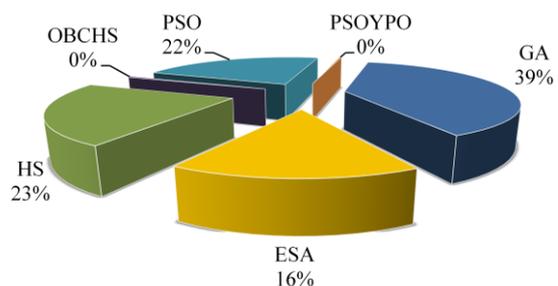


Figure 5. Comparison of STD results for TS14

From Table 8 and Table 9, it is realized that the chaos-embedded OBCHS can outperform the GA, ESA, HS and PSOYPO methods in dealing with TS1 to TS12. However, in many applications, the quality of solutions and efficiency of algorithms is more important than the running time. The UWL is not an exception. Figures 6 and 7 also compare the computational speed of different algorithms for NS and LS problems, respectively. From these illustrations, it can be understood that the proposed PSOYPO can show an acceptable performance in dealing with UWL problems.

ID	GA	ESA	HS	OBCHS	PSOYPO
TS1	0.287	0.041	0.042	0.014	0.552
TS2	0.322	0.028	0.210	0.021	1.127
TS3	0.773	0.031	0.215	0.028	2.451
TS4	0.200	0.018	0.041	0.014	4.445
TS5	0.801	0.256	0.704	0.098	5.412
TS6	0.896	0.098	0.085	0.019	6.102
TS7	1.371	0.119	1.012	0.064	7.921
TS8	0.514	0.026	0.201	0.014	8.930
TS9	6.663	2.506	4.122	0.514	9.754
TS10	5.274	0.446	0.492	0.197	9.851
TS11	7.189	0.443	0.785	0.312	10.821
TS12	2.573	0.079	1.415	0.142	10.711

Table 7. Comparison of computational performance for normal scale problems

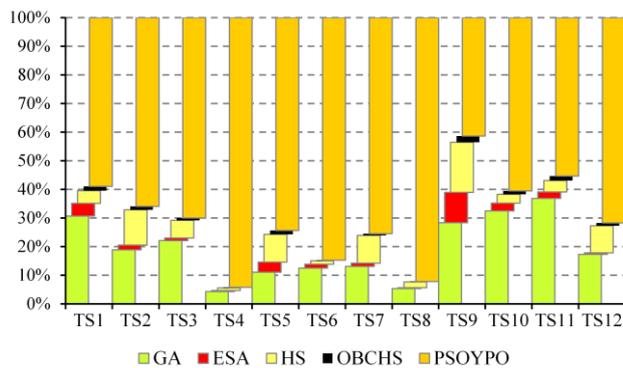


Figure 6. Comparison of time results for NS problems

ID	GA	ESA	HS	OBCHS	PSOYPO
TS13	184.422	17.930	47.412	15.741	19.851
TS14	510.445	91.937	147.52	84.112	124.874
TS15	591.516	131.345	110.430	101.922	352.811

Table 8. Comparison of computational performance for LS problems

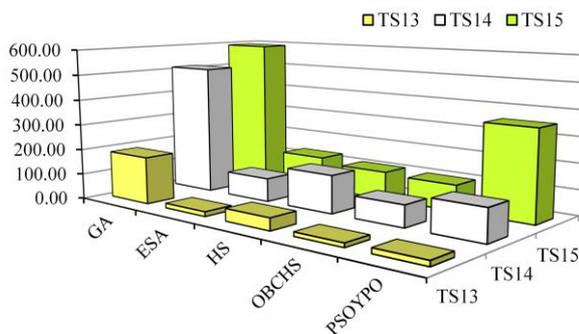


Figure 7. Comparison of time results for LS problems

To validate the efficacy of the PSOYPO, the Wilcoxon rank sum test with 0.05 degree is performed to finish the experiments. The rank sum test can demonstrate the significant improvements in the exploration and exploitation potentials of the proposed PSOYPO over other methods. The final results are tabulated in Table 9. In this Table, + shows that the improvements are significant and p-values are lower than 0.05. The negative values show that the PSOYPO cannot statistically outperform the second method.

Problem	PSOYPO versus				
ID	GA	ESA	HS	OBCHS	PSO
TS1	+	+	+	+	+
TS2	+	+	+	+	+
TS3	+	+	+	+	+
TS4	+	+	+	+	+
TS5	-	+	-	+	+
TS6	+	+	+	+	+
TS7	+	-	+	+	+
TS8	+	+	+	-	+
TS9	+	+	+	+	+
TS10	+	+	+	-	+
TS11	+	+	+	-	+
TS12	+	+	+	+	+

Table 9. Results of the statistical tests

From Table 9, it can be seen that the new results are significantly better than those of PSO algorithm in all cases. However, it is seen that the results are not statistically superior to the OBCHS in solving TS08, TS10, and TS11.

## 7. CONCLUSIONS AND FUTURE DIRECTIONS

In this paper, a new hierarchical PSO-based algorithm is proposed and employed to solve several UWL problems. The Yin-Yang-pair optimization (YYPO) was used to enrich the optimality of the exploited results. The YYPO is the latest MA proposed in 2015 inspiring from the philosophy of balance between conflicting concepts. The proposed PSOYPO was tested on 15 well-known small and large scale benchmark problems. It was observed that the PSOYPO can attain the best solutions throughout a reasonable computational time. The comparative results of this research vividly show that the immature convergence problems of PSO are alleviated, significantly. The New PSOYPO not only exploits better solutions but also it is capable of outperforming PSO in dealing with the UWL tasks.

In addition to utilizing the UWL models and the PSO and YYPO algorithms to real-world tasks, proposing more efficient and competent metaheuristics seems to be a remarkable research direction. Future works in this direction can revolve around the addition of more PSO operators to enhance its efficacy. Proposing new exploration or exploitation approaches can similarly deserve investigation. With adaptation, the PSOYPO technique employed in this paper can be extended to other FL scenarios, such as the capacitated FL (CFL) and the single source CFL problems.

## REFERENCES

Aardal, K., Chudak, F.A., Shmoys, D.B., 1999. A 3-approximation algorithm for the k-level uncapacitated facility location problem. *Information Processing Letters* 72, pp. 161-167.

- Al-Sultan, K., Al-Fawzan, M., 1999. A tabu search approach to the uncapacitated facility location problem. *Annals of Operations Research* 86, pp. 91-103.
- Aydin, M.E., Fogarty, T.C., 2004. A distributed evolutionary simulated annealing algorithm for combinatorial optimisation problems. *Journal of heuristics* 10, pp. 269-292.
- Azad, M., Kalam, A., Rocha, A.M.A., Fernandes, E.M.d.G., 2013. A simplified binary artificial fish swarm algorithm for uncapacitated facility location problems, *World Congress on Engineering 2013, WCE 2013*. Newswood Limited Publisher, pp. 31-36.
- Barcelo, J., Hallefjord, Å., Fernandez, E., Jörnsten, K., 1990. Lagrangean relaxation and constraint generation procedures for capacitated plant location problems with single sourcing. *OR Spectrum* 12, pp. 79-88.
- Basti, M., Sevkli, M., 2015. An artificial bee colony algorithm for the p-median facility location problem. *International Journal of Metaheuristics* 4, pp. 91-113.
- Basu, S., Sharma, M., Ghosh, P.S., 2015. Metaheuristic applications on discrete facility location problems: a survey. *Opsearch* 52, pp. 530-561.
- Erlenkotter, D., 1978. A dual-based procedure for uncapacitated facility location. *Operations Research* 26, pp. 992-1009.
- Esnaf, Ş., Küçükdeniz, T., Tunçbilek, N., 2014. Fuzzy C-Means Algorithm with Fixed Cluster Centers for Uncapacitated Facility Location Problems: Turkish Case Study, *Supply Chain Management Under Fuzziness*. Springer, pp. 489-516.
- Ghosh, D., 2003. Neighborhood search heuristics for the uncapacitated facility location problem. *European Journal of Operational Research* 150, pp. 150-162.
- Guner, A.R., Sevkli, M., 2008. A discrete particle swarm optimization algorithm for uncapacitated facility location problem. *Journal of Artificial Evolution and Applications* 2008, pp. 1-9.
- Guo, P., Cheng, W., Wang, Y., 2017. Hybrid evolutionary algorithm with extreme machine learning fitness function evaluation for two-stage capacitated facility location problems. *Expert Systems with Applications* 71, pp. 57-68.
- Heidari, A.A., Abbaspour, R.A., Jordehi, A.R., 2017a. An efficient chaotic water cycle algorithm for optimization tasks. *Neural Computing and Applications* 28, pp. 57-85.
- Heidari, A.A., Abbaspour, R.A., Jordehi, A.R., 2017b. Gaussian bare-bones water cycle algorithm for optimal reactive power dispatch in electrical power systems. *Applied Soft Computing* 57, pp. 657-671.
- Heidari, A. A., Pahlavani, P., 2017c. An Efficient Modified Grey Wolf Optimizer with Lévy Flight for Optimization Tasks. *Applied Soft Computing*, 60, pp. 115-134.
- Heidari, A.A., Ali Abbaspour, R., 2016. Let a CPSO+ algorithm explore your high-quality shortest paths: An effective chaos-enhanced PSO-based strategy, *Proceedings of the International Conference on Industrial Engineering and Operations Management*, pp. 2125-2132.
- Heidari, A.A., Delavar, M.R., 2016. A Modified Genetic Algorithm for Finding Fuzzy Shortest Paths in Uncertain Networks. *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.* XLI-B2, pp. 299-304.
- Heidari, A.A., Kazemizade, O., Abbaspour, R.A., 2015a. OBCHS: An Effective Harmony Search Algorithm with Opposition based Chaos-enhanced Initialization for Solving Uncapacitated Facility Location Problems. *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.* XL-1/W5, pp. 307-311.
- Heidari, A.A., Mirvahabi, S.S., Homayouni, S., 2015b. An Effective Hybrid Support Vector Regression with Chaos-Embedded Biogeography-Based Optimization Strategy for Prediction of Earthquake-Triggered Slope Deformations. *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.* XL-1/W5, pp. 301-305.
- Jaramillo, J.H., Bhadury, J., Batta, R., 2002. On the use of genetic algorithms to solve location problems. *Computers & Operations Research* 29, pp. 761-779.
- Klose, A., 1998. A branch and bound algorithm for an uncapacitated facility location problem with a side constraint. *International Transactions in Operational Research* 5, pp. 155-168.
- Körkel, M., 1989. On the exact solution of large-scale simple plant location problems. *European Journal of Operational Research* 39, pp. 157-173.
- Korupolu, M.R., Plaxton, C.G., Rajaraman, R., 2000. Analysis of a local search heuristic for facility location problems. *Journal of algorithms* 37, pp. 146-188.
- Michel, L., Van Hentenryck, P., 2004. A simple tabu search for warehouse location. *European Journal of Operational Research* 157, pp. 576-591.
- Ng, T.L., 2014. Expanding Neighborhood Tabu Search for facility location problems in water infrastructure planning, 2014 IEEE International Conference on Systems, Man and Cybernetics (SMC). IEEE, pp. 3851-3854.
- Ou-Yang, C., Ansari, R., 2017. Applying a hybrid particle swarm optimization\_Tabu search algorithm to a facility location case in Jakarta. *Journal of Industrial and Production Engineering* 34, pp. 199-212.
- Punnathanam, V., Kotecha, P., 2017. Multi-objective optimization of Stirling engine systems using Front-based Yin-Yang-Pair Optimization. *Energy Conversion and Management* 133, pp. 332-348.
- Saha, S., Kole, A., Dey, K., 2011. A Modified Continuous Particle Swarm Optimization Algorithm for Uncapacitated Facility Location Problem, *Information Technology and Mobile Communication*. Springer, pp. 305-311.
- Sevkli, M., Guner, A.R., 2006. A continuous particle swarm optimization algorithm for uncapacitated facility location problem, *International Workshop on Ant Colony Optimization and Swarm Intelligence*. Springer, pp. 316-323.
- Sun, M., 2006. Solving the uncapacitated facility location problem using tabu search. *Computers & Operations Research* 33, pp. 2563-2589.
- Tcha, D.-w., Lee, B.-i., 1984. A branch-and-bound algorithm for the multi-level uncapacitated facility location problem. *European Journal of Operational Research* 18, pp. 35-43.
- Trelea, I.C., 2003. The particle swarm optimization algorithm: convergence analysis and parameter selection. *Information processing letters* 85, pp. 317-325.