HIDMS-PSO: A New Heterogeneous Improved Dynamic Multi-Swarm PSO Algorithm

Fevzi Tugrul Varna
Department of Informatics, University of Sussex
Brighton, United Kingdom
f.varna@sussex.ac.uk

Phil Husbands
Department of Informatics, University of Sussex
Brighton, United Kingdom
philh@sussex.ac.uk

***Abstract*—In this paper, a variant of the particle swarm optimisation (PSO) algorithm is introduced with heterogeneous behaviour and a new dynamic multi-swarm topological structure. The new topological structure enables the algorithm to have more control over the interaction and information exchange between the particles to reduce the loss of diversity and avoid premature convergence. In the new algorithm, the population is initially divided into two sub-populations, first sub-population is further divided into sub-swarms that are formed using the introduced topological structure. The particles of sub-swarms are guided using heterogeneous behaviour by selecting various exemplars. The second sub-population employs the classical PSO search with local and global information to simulate a homogenous behaviour. There is information flow between the two sub-populations. The algorithm was tested on the CEC2005 and CEC2017 test suites with comparison against various state-of-the-art PSO variants and other state-of-the-art meta-heuristics. The experimental results show that for the two test suites, the proposed algorithm outperformed the majority of the state-of-the-art algorithms on most problems.**

***Index Terms*—particle swarm optimisation, swarm intelligence, meta-heuristics**

I. Introduction

The particle swarm optimisation algorithm, introduced by Kennedy and Eberhard in 1995 [1] [2], is a search algorithm designed to solve single-objective optimisation problems. The PSO consists of a collection of agents, referred to as particles that each represent a candidate solution for the given problem. Due to its simplicity, the PSO algorithm has attracted the interest of many researchers over the past few decades. As a result, PSO became one of the predominant swarm algorithms applied to various problems including task allocation [3], image processing [4], feature selection [5] and robotic applications [6]. However, the standard PSO algorithm's capability to solve certain kinds of complex problems (e.g. high-dimensional or with many local optima) is limited. The lack of success in more complex problems is generally tied to two main drawbacks of the PSO, namely loss of diversity and premature convergence [7]. These two drawbacks are related to one another as loss of diversity in a population diminishes the swarm's capability to focus on other solutions or attract particles away from the local solution, hence resulting in premature convergence. Because of the aforementioned issues, numerous variants of the PSO were introduced to improve the canonical algorithm and solve problems in different domains. In short, the focus of improvements of PSO variants can be divided into four categories, namely: parameter tuning, neighbourhood topology, learning strategy and hybridisation with other methods. Shi and Eberhart [8] proposed a linearly decreasing inertia weight parameter to control the balance between exploration and exploitation, which became one of the most widely used control mechanism for the ω (inertia weight) parameter. [9] introduced the use of constriction coefficients to guarantee the convergence of the PSO, [10] introduced time-varying c_1 and c_2 parameters and [11] proposed a fuzzy logic approach to determine the inertia weight, acceleration coefficients and the clamping values for velocity independently for each particle. The neighbourhood topology of the PSO algorithm can change the interaction between particles and may have a significant impact on the swarm's exploration and exploitation behaviour. Generally, the neighbourhood topology structure can be categorised as either a static or dynamic topology. The classic static topology structure is g_{best} (global best) and l_{best} local best. [12] introduced a variant of a PSO and studied various topological structures including square structures, four clusters and pyramids. The study [13] introduced a variant of PSO with a dynamic multi-swarm topological structure that divides the overall swarm into smaller sub-swarms. Similarly, [14] proposed a multi-swarm PSO with mixed search behaviour to maintain swarm diversity and [15] introduced increasing topology connectivity to enhance the control of exploration and exploitation. This paper presents a novel variant of the PSO, HIDMS-PSO, which builds on and improves the DMS-PSO [16] by introducing a new topological structure for small-subswarms and learning exemplars to enhance the search behaviour, and reduce loss of diversity and premature convergence. The study [38] also employs the concept of master-slave. It uses the symbiotic relationship between the master and slave at swarm level to balance the exploration and exploitation where a population consists of one master swarm and several slave swarms. The study [39] extends SRPSO and employs a directional update strategy and uses the median of the personal best of random particles as the personal best for the poorly performing particles. In our approach, a portion of the population is assigned master-slave roles at an individual level with a specific topological structure (detailed in section 3), and further hierarchical ranks assigned

978-1-7281-2547-3/20/$31.00 ©2020 IEEE

to slave particles (as shown in Fig. 2 and Fig. 3) in each unit. The topological structure is supported by the communication model to restrict arbitrary communication among particles to allow hierarchal, one and two-way communication within and among other particles to avoid particles focusing on a single best-solution. In the present study, two movement strategies are introduced, namely, inward and outward strategy (detailed in section 3). The inward strategy guides particles towards a local solution while the outward strategy guides particles away from the self's unit. Both strategies combined allows the exploitation of the local solution and exploration of solutions discovered by other units. Also, the master-slave roles, the communication model and the heterogeneous behaviour of particles further assist escaping from local optima and maintaining population diversity for extended periods. Comparative analysis on demanding test suites shows that the proposed algorithm HIDMS-PSO is able to outperform state-of-the-art meta-heuristics and PSO variants.

II. Related Studies

This section provides the necessary background information about the canonical PSO and DMS-PSO algorithms.

A. Canonical PSO

In the canonical PSO, particles are initially randomly distributed in the search space. Throughout the search process, particles learn and retain certain information about the environment, namely its position, velocity and personal best position found. At each iteration, the position of the particle is updated by adding together its current position and velocity. The velocity has the most significant influence on the next position of the particle, and is calculated using two pieces of information: namely the particle's personal best-known position and the best position found within the swarm. The velocity and position calculation of the canonical PSO is as follows:

$$v_i^{(t+1)} = \omega v_i^{(t)} + c_1 r_1 \left(p_{best} - x_i^{(t)}\right) + c_2 r_2 \left(g_{best} - x_i^{(t)}\right) \tag{1}$$

$$x_i^{(t+1)} = x_i^{(t)} + v_i^{(t)} \tag{2}$$

Where ω, c_1 and c_2 are control parameters, namely the inertia weight and acceleration coefficients, $v_i^{(t)}$ is the i^{th} particle's velocity, p_{best} is the personal best position, g_{best} is the globally best known solution and $x_i^{(t)}$ is the current position of the i^{th} particle. Here, r_1 and r_2 are random variables in the range of [0,1].

B. DMS-PSO

DMS-PSO [16] is a variant of the PSO algorithm with a dynamic topological structure. DMS-PSO searches by initially segregating the population into many relatively small sub-swarms and each sub-swarm searches based on their best historical information with no communication with other sub-swarms during the search; essentially a co-evolutionary PSO with sub-swarms searching in parallel. PSO's general tendency to converge rapidly result in sub-swarms converging to local optima. To overcome this issue, DMS-PSO employs a regrouping strategy that takes place at specific intervals to continue searching using a new configuration of small swarms, and by employing a regrouping strategy, viable information is then exchanged between the swarms at every R generations. Fig. 1 shows the search phases of the DMS-PSO algorithm, assuming a population of 9 particles, randomly divided into 3 sub-swarms. Subsequently, each particle searches within its sub-swarm to find a better solution. The position of particles for the DMS-PSO algorithm is updated using Eqs. (2) and (3).

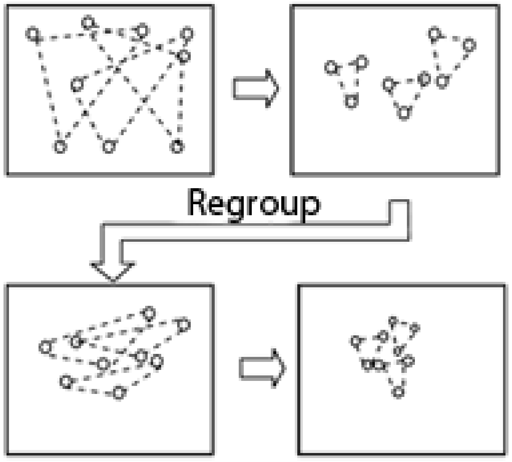

Fig. 1. Search phases of the DMS-PSO algorithm.

$$v_i^{(t+1)} = \omega v_i^{(t)} + c_1 r_1 \left(p_{best} - x_i^{(t)}\right) + c_2 r_2 \left(l_{best} - x_i^{(t)}\right) \tag{3}$$

III. Proposed Method: HIDMS-PSO

The present study aims to improve the performance of the DMS-PSO algorithm by introducing a new dynamic topological structure and heterogeneous particle behaviour. The homogenous and heterogenous sub-populations allocated enhance the exploration and exploitation capabilities without impeding one another.

1) Proposed Topological Structure: The new structure is composed of sub-swarm like entities called units. Each unit has a fixed population of 4 particles, with a single master particle (p_{master}) selected randomly and 3 slave particles with distinct types (p^1_{slave})...(p^3_{slave}). Master and slave particles retain their roles throughout the search process. The distinction in type between the slave particles is intended to allow heterogeneous behaviour, restrict information flow to avoid premature convergence and depletion of diversity. Fig. 2 exhibits the structure of a single unit.

2) Communication Model: Information flow and interaction between particles play a significant role in maintaining the swarm diversity and particle's guidance. In the present study, instead of a global or arbitrary exchange of information, the

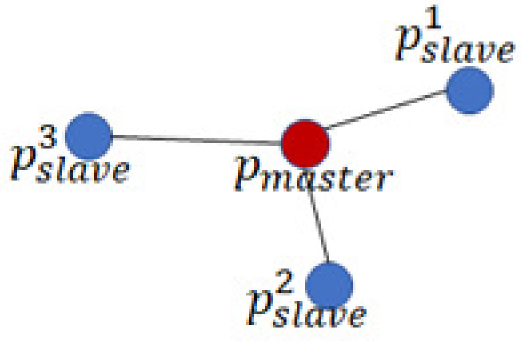

Fig. 2. Topological structure of a single unit.

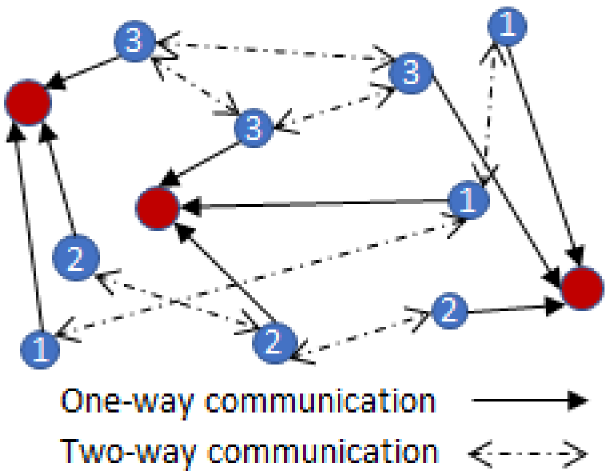

Fig. 3. The visual depiction of the communication model between 3 units.

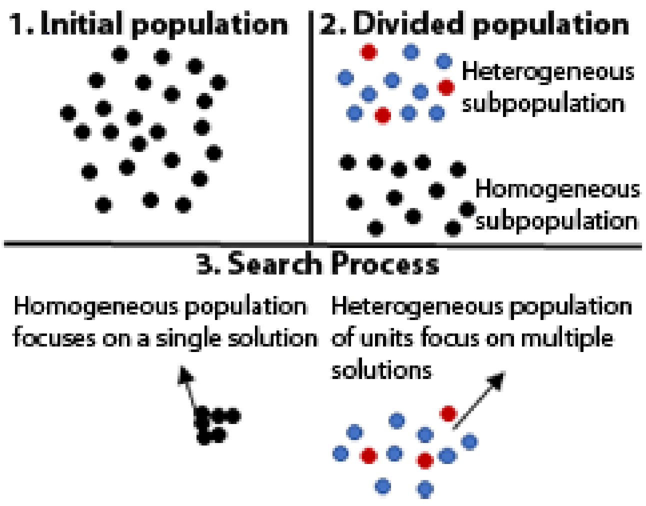
1. Initial population
2. Divided population
Heterogeneous subpopulation
Homogeneous subpopulation
3. Search Process
Homogeneous population focuses on a single solution
Heterogeneous population of units focus on multiple solutions

Fig. 4. Search phases of the HIDMS-PSO algorithm.

unit structure hierarchy between particles and the assumed distinction between the slave particles is used to restrict and control the flow information between the particles. The communication model proposed in this study can be summarised using the following rules:

1) Particles of the ith unit do not directly communicate with the particles of the jth unit. Communication is established via the slave particles only.
2) Master particles can only exchange information with one of their slaves.
3) Slave particles can only communicate with the slaves of the same type; hence they cannot communicate with the slaves within their unit.

3) Search Behaviour: The search process of the proposed algorithm initiates by separating the population into two equal sub-populations as heterogenous and homogenous sub-populations. The first segment of the population is used to form N of the units shown in Fig. 2. Each unit exhibits heterogeneous behaviour (detailed later) while the second sub-population exhibits homogeneous behaviour by conducting a PSO search using the update equations 1 and 2. The sub-population dedicated to forming units exhibits two different search behaviour using an inward-oriented strategy and an outward-oriented strategy. The inward-oriented behaviour emphasis is on guiding particles' movement using the information obtained from members of the unit the particle belongs to. On the contrary, the outward-oriented behaviour guides particles based on the information obtained from other units. In the proposed algorithm, the particle of the heterogenous sub-population randomly selects one of the strategies, although a more precise method may be designed to improve switching between the two strategies. It's worth noting that the second sub-population is completely segregated from the units and performs an independent search based on classical PSO update equations while the first segment of the population (units) employs heterogeneous search behaviour. For both movement strategies briefly mentioned above, the selection of an exemplar is determined based on a single factor, that is the type of the particle (i.e. master, slave or slave type).

a) Inward-oriented strategy: The inward-oriented strategy uses information from members of the unit to guide its members. For master particles of the Nth unit, this strategy allows particles to update their velocities by randomly selecting one of the equations Eqs. 4-6:

$$v_m^{(t+1)} = \omega(t)\, v_m^{(t)} + c_1 r_1 \left(p_{best} - x_m^{(t)}\right) + c_2 r_2 \left(x_s^{dis} - x_m^{(t)}\right) \tag{4}$$

Where x_s^{dis} is the most dissimilar slave particle (positional dissimilarity) in the unit N. Movement towards the most dissimilar slave particle boosts the diversity of the master particle, hence the whole unit, as slave particles of a unit are highly influenced by the master particle's position. The global p_{best} can come from either of the two main sub-populations.

$$v_m^{(t+1)} = \omega\left(t\right) v_m^{(t)} + c_1 r_1 \left(p_{best} - x_m^{(t)}\right) + c_2 r_2 \left(x_s^{best} - x_m^{(t)}\right) \tag{5}$$

Where x_s^{best} is the slave particle with the lowest cost in unit N. Local exploration is performed by guiding the master particle towards the best slave particle.

$$v_m^{(t+1)} = \omega(t)\, v_m^{(t)} + c_1 r_1 \left(p_{best} - x_m^{(t)}\right) + c_2 r_2 \left(x_s^{avg} - x_m^{(t)}\right) \tag{6}$$

On the contrary, for the slave particles, the only option provided for this strategy is to move towards the unit master and p_{best}, as shown in Eq. 7.

$$v_s^{(t+1)} = \omega(t)\, v_i^{(t)} + c_1 r_1 \left(p_{best} - x_s^{(t)}\right) + c_2 r_2 \left(x_m - x_s^{(t)}\right) \tag{7}$$

Where x_m is the master particle of the N^{th} unit.

b) Outward-oriented strategy: As oppose to inward-oriented movement, outward-oriented movement allows particles to learn from other units while maintaining hierarchical structure. The master particle can randomly select one of the following equations to guide its behaviour:

$$v_m^{(t+1)} = \omega(t)\, v_m^{(t)} + c_1 r_1 \left(p_{best} - x_m^{(t)}\right) + c_2 r_2 \left(x_{unit}^{avg} - x_m^{(t)}\right) \tag{8}$$

Where x_{unit}^{avg} is the average position of the N unit's particles.

$$v_m^{(t+1)} = \omega(t)\, v_m^{(t)} + c_1 r_1 \left(p_{best} - x_m^{(t)}\right) + c_2 r_2 \left(x_{unit}^{m} - x_m^{(t)}\right) \tag{9}$$

Where x_{unit}^{m} is the position of the master of the N unit.

$$v_m^{(t+1)} = \omega(t) v_m^{(t)} + c_1 r_1 \left(x^{avg} - x_m^{(t)}\right) + c_2 r_2 \left(x_{unit}^{m} - x_m^{(t)}\right) \tag{10}$$

Where x^{avg} is the average position of particle's own unit members and x_{unit}^{master} is the position of the master particles of the N unit. Similar to the slave particle's movement in the inward-oriented strategy, in this case, the slave particles employ a single update equation to move towards a random slave of the same type that belongs to another unit, using:

$$v_s^{(t+1)} = \omega\left(t\right) v_s^{(t)} + c_1 r_1 \left(p_{best} - x_s^{(t)}\right) + c_2 r_2 \left(x_{unit}^{m} - x_s^{(t)}\right) \tag{11}$$

By combining both strategies and allowing a portion of the population to employ classical position update using Eqs .1 and 2, the overall swarm is divided into homogenous (first segment) and heterogeneous (second segment) groups and the heterogeneous population is further divided into N number units for efficient exploration and exploitation. The exemplars employed in the inward-oriented strategy allows master particles to perform exploration based on several different guidances and the slave particles are guided to exploit the master's best-known position. This strategy essentially allows each unit to explore and exploit local solutions simultaneously. Similarly, the outward-oriented strategy facilitates particles of a unit to learn from other units' members and the overall knowledge which results maintains their diversity at a certain level and allows members of units to explore more promising regions at time t to escape from potential local optima. As a result, the combination of homogenous and heterogeneous populations favours maintaining the balance of exploration and exploitation while inward and outward learning strategies allow particles to initiate single-timed fluctuations in their behaviour that enhances individual unit's diversity and escape from local minima. [17] introduced a non-uniform mutation operator to further improve the loss of population diversity and enhance the exploration capability in the swarm. The present study employs a similar approach by introducing a partial mutation on particles using a non-uniform mutation with relatively small fixed mutation probability of Pm=0.1. The mutation takes place on different dimensions of each particle selected randomly at specific intervals to search tiny areas surrounding particles' position. The number of dimensions and which dimensions to mutate are determined randomly for each particle at specific intervals (see the pseudocode).

c) Parameters: As mentioned in the previous sections, the PSO algorithm has 3 parameters namely, acceleration coefficients c_1, c_2, and inertia weight ω. In the literature, many studies propose optimum parameter settings for different problem sets and control methods are introduced to maintain the optimum control of the exploration and exploitation balance. In the present study, c_1 and c_2 are set as time-varying acceleration coefficients introduced in [26] which improves the global exploration during the early stage and encourages convergence during later stages of the search process. The inertia weight parameter is controlled by nonlinearly decreasing using the sigmoid function [17]. The sigmoid function's smooth, monotone and continuous properties facilitate a good balance of linearity and non-linearity [17]. Similar to the regrouping strategy employed in the DMS-PSO, the present study employs a related parameter. However, the introduced mechanism does not intent to regroup and exchange information but rather reshape the units while maintaining the master-slave structure. The intention is to further improve the diversity of the smaller groups within the overall swarm by stochastically switching particles of each unit while maintaining their hierarchies.

IV. Experimental Results and Discussions

This section presents the experimental design and results. The first subsection describes the experimental setup, benchmark suites, and statistical analysis and the latter presents the results of the two experiments conducted on the CEC2005 and CEC2017 benchmark suites.

1) Experimental Setup: The present study conducted two experiments to examine the performance of the proposed method, namely using the CEC2005 and CEC2017 benchmark test suites. In the first experiment, the proposed method was compared with 9 state-of-the-art PSO variants, namely, inertia weight PSO [34], HCLDMS-PSO [17], FDR-PSO [28], DMS-PSO [16], HPSO-TVAC [29], MNHPSO-JTVAC [30], CLPSO [31], SRPSO [32] and HCLPSO [33] on the CEC2005 test suite. The population size was set to 40 for all methods [17]. In the second experiment, the proposed method was compared with the results of 2 inertia weight PSO algorithms with different parametric settings, state-of-the-art DMS-PSO and other state-of-the-art meta-heuristics (including the bat algorithm (BA) [18], grey wolf optimizer (GWO) [19], butterfly optimization algorithm (BOA) [20], whale optimization algorithm (WOA) [21], moth flame optimization (MFO) [22], artificial bee colony (ABC) [24], flower pollination algorithm (FPA) [27], cuckoo search algorithm (CS) [23] and invasive weed optimisation (IWO) [37] algorithm on the CEC2017 special session on real-parameter single objective optimisation benchmark suite [25]. The CEC2017 test suite consists of 30 test functions, and for all problems, the lower and upper bound range was $[-100, 100]^d$. The population size of the PSO[1], PSO[2], DMS-PSO and the proposed method was set to 40. The population size for the other nine meta-heuristics was set to 100 [17]. For detailed parameter values on the comparative methods and details of the test suites, refer to [17] [32] and the original studies. For both experiments, each problem was tested 30 times, 300,000 function evaluations for 30 dimensional and 500,000 function evaluations for 50-dimensional problems on both test suites. Table I and Table II displays the mean errors obtained for the CEC2005 test suite for 30 and 50 dimensional problems and Table III and Table IV displays the mean errors obtained for the CEC2017 test suite for 30 and 50 dimensional problems. Table V and Table VI shows the average and final ranks of the mean performances for both test suites. The Wilcoxon signed-rank test conducted on the final ranks obtained for the CEC2005 problems reveals that the result is significant between all comparison methods and the proposed algorithm except HCLDMS-PSO and HCLPSO at $p < 0.05$ for problem size of 30 and 50 dimensions. The Wilcoxon signed-rank test conducted on the final ranks obtained for the CEC2017 problems reveals that the result is significant between the proposed algorithm and all comparison methods except DMS-PSO for problem size of 30 dimensions. The result is significant between the proposed algorithm and all comparison methods for a problem size of 50 dimensions at $p < 0.05$. Due to length restriction of this paper, experimental results are partially included. An external supplementary material is provided for complete results of experiments that can be accessed from users.sussex.ac.uk/fv47/HIDMSPSO.pdf.

A. Results

Experimental results for the CEC2005 test suite for problems of 30 dimensions reveal that for problems F1-F7, F11, F12, F14, F18-20 and F22, the SRPSO acquired the best performance. For these problems, the second-best performances were attained (in problem order) by HPSO-TVAC, HIDMS-PSO, MNHPSO-JTVAC, DMS-PSO, HCLPSO, HCLDMS-PSO, HIDMS-PSO, DMS-PSO, HIDMS-PSO, HCLDMS-PSO (for F18-F20) and HCLPSO. For problem F8, HIDMS-PSO achieved the best, and the DMS-PSO achieved the second-best performance. CLPSO exhibited the best performance for problems F9 and F15 followed by HCLPSO for the second-best performance, and CLPSO attained the best performance for F13, whereas the SRPSO exhibited the second-best performance. For problems F10, F17, F21 and F24, the HCLDMS-PSO showed the best mean performance, for the same problem set the second-best performance was achieved by SRPSO, HCLPSO, CLPSO and MNHPSO-JTVAC. The second experiment conducted on the CEC2005 test suite for the problem size of 50 dimensions reveals that the SRPSO algorithm exhibited a similar performance pattern by achieving the best performance for problems F1-F7, F11, F12, F14, F18-F20, F22-F24. For the same problem set, the second best performance was achieved by (in the same order) HPSO-TVAC, HIDMS-PSO, MNHPSO-JTVAC, DMS-PSO (for F4 and F5), HIDMS-PSO (for F6 and F7), DMS-PSO, HIDMS-PSO, HCLDMS-PSO (for F14 and F18), HCLPSO (for F19, F20 and F22), HIDMS-PSO and HCLDMS-PSO. For problems F9, F15 and F21, CLPSO exhibited the best mean performance followed by HCLPSO and HCLDMS-PSO for the second-best performance. HCLPSO attained the best performance for problems F13, F17 and F25 and SRPSO and HIDMS-PSO exhibited the second-best performance for the same problem set. For problems F10 and F16, HCLDMS-PSO showed the best mean performance while the SRPSO attains the second-best performance in both cases. For problem F8, HIDMS-PSO outperformed comparison methods, and the DMS-PSO achieved the second-best performance. Table I and Table II display the results for the CEC2005 experiment, and Table V shows the average and final ranks for this experiment. For the problem size of 30 dimensions, the SRPSO attained the top rank, followed by the second rank achieved by the HCLDMS-PSO algorithm. The proposed method achieved the 3rd rank among the nine state-of-the-art PSO variants by outranking (in their rank order) HCLPSO, DMS-PSO, MNHPSO-JTVAC, HPSO-TVAC, CLPSO, PSO and FDR-PSO. For the problem size of 50 dimensions, again, the top rank is attained by SRPSO followed by the second rank achieved by the HIDMS-PSO. Experimental results for the CEC017 test suite for the size of 30 dimensions reveal that for problems F3, F7, F8, F12 and F21, HIDMS-PSO achieved the best performance. For the same problem set, DMS-PSO attained the second-best performance (for problems F3, F7, F8) and the CS algorithm (for F12 and F21). For problems F5, F9, F11, F16, F17, F20, F22-F24 and F27-F30, DMS-PSO exhibited the best mean performance, and HIDMS-PSO achieved the second-best performance for all problems except F30, where ABC attained the second-best performance instead. For problems F1, F4, F10, F25 and F26 ABC exhibited the best mean performance, for the same problem set the second-best performance is observed by DMS-PSO, HIDMS-PSO (for F4, F10, F26) and CS. The CS algorithm outperformed comparison methods for problems F13-F15, F18 and F19. The ABC algorithm achieved the second-best performance (for F13, F15 and F19) and HIDMS-PSO (for F14 and F18). For problem F6, the HIDMS-PSO, DMS-PSO and ABC algorithms achieved the same best performance, and the GWO algorithms attained the second-best performance. The second experiment conducted on the CEC2017 test suite for the problem size of 50 dimensions reveal that HIDMS-PSO achieved the best mean performance for problems F3, F4, F5, F6, F7, F8, F9, F11, F12, F21, F23, F24 and F30, for the same problem set the second-best performance is attained by DMS-PSO (for F3, F5, F6, F7, F8, F9, F21, F23, F24 and F30), and CS (for F4, F11 and F12). The CS algorithm attained the best mean performance for problems F13-15, F18 and F19. For the same problem set, the second-best performance is observed by HIDMS-PSO (for F13, F15 and F18), (for F14 and F19) IWO and DMS-PSO. For problems F25 and F28, the CS algorithm, for F26 and F29 DMS-PSO and problem F27, the ABC algorithm exhibited the best performance. HIDMS-PSO achieved the second-best performance for the same problem set. The DMS-PSO algorithm attained the best mean performance for problems F1, F10, F16, F17, F20 and F22 and HIDMS-PSO achieved the second-best performance for problems F1, F16 F20, F22, along with GWO for problems F10 and F17. Table III and Table IV display the results for the CEC2017 experiment, and Table VI shows the average and final ranks for this experiment. Calculation of the average and final ranks indicate that the proposed algorithm, HIDMS-PSO, achieved the best performance and the top rank for the CEC2017 test suite for a problem size of 30 and 50 dimensions and the DMS-PSO algorithm ranks second for both 30 and 50-dimensional problems.

V. Conclusions

The present study proposed the HIDMS-PSO algorithm, a novel variant of PSO with a new topological structure. The algorithm divides the population to establish a heterogeneous and a homogenous population with distinct behaviours. The new topological structure is used to control the interaction and the flow of information between the particles. The units and the topological structure combined significantly contributed to guiding the particles by either using inward or outward-oriented strategies to perform better exploration/exploitation, and to improve loss of diversity and premature convergence. The algorithm was tested on the CEC2005 and CEC2017 test suites at 30 and 50 dimensions with state-of-the-art PSO variants and meta-heuristics. The proposed algorithm's comparative performance was very strong on both test suites, including being ranked top for the CEC2017 suites at both 30 and 50 dimensions, indicating its potential as a powerful search method. The present work can be extended by applying the HIDMS-PSO algorithm to complex practical problems, or it can be further improved to compete with more complex algorithms.

References

[1] R. C. Eberhart, and J. Kennedy, "A new optimizer using particle swarm theory", Proc. of the Sixth Int. Symposium on Micromachine and Human Science, Nagoya, Japan. pp. 39-43, 1995

[2] J. Kennedy, and R. C. Eberhart, "Particle swarm optimization ". Proceedings of IEEE International Conference on Neural Networks, Piscataway, NJ. pp. 1942-1948, 1995

[3] Li, M., Liu, C., Li, K., Liao, X. and Li, K., 2020. Multi-task allocation with an optimized quantum particle swarm method. Applied Soft Computing, 96, p.106603.

[4] Santos Junior, J. and Silva do Monte Lima, J., 2018. Particle swarm optimization for 3D object tracking in RGB-D images. Computers and Graphics, 76, pp.167-180.

Algorithm 1: HIDMS-PSO

```
population size $n$, $C$=0.15, $w_{max}$=0.99, $w_{min}$=0.2;
$c_1 = 2.5 - (1 : T_{max} * 2/T_{max})$;
$c_2 = 0.5 - (1 : T_{max} * 2/T_{max})$;
$w_1\,(t) = \frac{w_{max}+(w_{min}-w_{max})}{1+exp\left(-5\left(\frac{2t}{T_{max}}-1\right)\right)}$;
for $t$=1:$T_{max}$ do
    for $i$=1:$n$ do
        if mod(t,$T_{max}$*0.1)==0 then
            vertically shuffle slave particles
        end
        if mod(t,$T_{max}$*0.05)==0 then
            if $t < T_{max}$ then
                $\beta$ = round(d*U(0.1,1));
            else
                $\beta$ = round(d*0.1);
            end
            for $j$=1:$n$ do
                select $\beta$ number of random dimensions to mutate
                  for each particle
            end
        end
        if $f(x_i) >= avg$ then
            $w$ = $w_1(t)$ + $C$; if $w >$ 0.99, $w$ = 0.99 end;
        else
            $w$ = $w_1(t)$ - $C$; if $w <$ 0.20, $w$ = 0.20 end;
        end
        if randi([0 1])==0 (inward-strategy) then
            if $i^{th}$ particle is a master then
                behaviour = randi([1 3]);
                if behaviour == 1 then
                    update $v_i$ and $x_i$ using Eqs. 4 and 2
                end
                if behaviour == 2 then
                    update $v_i$ and $x_i$ using Eqs. 5 and 2
                end
                if behaviour == 3 then
                    update $v_i$ and $x_i$ using Eqs. 6 and 2
                end
            else
                update $v_i$,$x_i$ using Eqs. 7 and 2
            end
        else
            if $i^{th}$ particle is a master then
                behaviour = randi([1 3]);
                if behaviour == 1 then
                    update $v_i$,$x_i$ using Eqs. 8 and 2
                end
                if behaviour == 2 then
                    update $v_i$,$x_i$ using Eqs. 9 and 2
                end
                if behaviour == 3 then
                    update $v_i$,$x_i$ using Eqs. 10 and 2
                end
            else
                update $v_i$,$x_i$ using the Eqs. 11 and 2
            end
        end
        perform partial non-uniform mutation on the $x_i$
        Evaluate the fitness of $x_i$
        Update the $p_{best}$ and $g_{best}$
    end
end
```

TABLE I
The mean error results obtained in CEC2005 test suite for problem size of 30 dimensions.

	HCLDMS-PSO	SRPSO	HCLPSO	MNHPSO-JTVAC	CLPSO	DMS-PSO	HPSO-TVAC	FDR	PSO	HIDMS-PSO*
F1	3.3E-12	0.0E+00	1.3E+01	5.9E-14	2.9E-06	2.6E+01	5.5E-14	5.0E+02	4.7E+03	1.4E-12
F2	3.5E+01	0.0E+00	2.2E+01	9.3E-03	8.2E+03	5.0E+01	4.8E-02	1.4E+03	6.5E+03	1.1E-03
F3	2.9E+06	4.5E+01	3.7E+06	9.8E+05	3.5E+07	3.9E+06	1.7E+06	1.6E+07	2.8E+07	1.1E+06
F4	2.2E+03	3.0E+01	2.1E+03	3.6E+03	1.9E+04	3.3E+02	3.0E+03	2.8E+03	8.4E+03	1.7E+03
F5	2.8E+03	1.8E+03	2.4E+03	5.4E+03	5.8E+03	3.1E+03	5.5E+03	3.6E+03	1.0E+04	3.0E+03
F6	6.3E+01	4.0E+01	2.9E+05	9.9E+01	2.5E+02	2.0E+05	1.1E+02	2.4E+06	5.6E+08	7.0E+01
F7	4.7E+03	7.6E-02	4.7E+03	4.7E+03	4.7E+03	4.7E+03	4.7E+03	4.7E+03	5.5E+03	4.7E+03
F8	2.1E+01	2.1E+01	2.1E+01	2.1E+01	2.1E+01	2.1E+01	2.1E+01	2.1E+01	2.1E+01	2.1E+01
F9	3.7E+01	3.8E+01	4.0E+00	2.5E+01	6.3E-04	7.2E+01	3.6E+01	2.7E+02	8.5E+01	5.0E+01
F10	3.5E+01	5.2E+01	6.7E+01	1.0E+02	1.8E+02	6.3E+01	1.0E+02	2.0E+02	1.4E+02	6.5E+01

TABLE II
The mean error results obtained in CEC2005 test suite for problem size of 50 dimensions.

	HCLDMS-PSO	SRPSO	HCLPSO	MNHPSO-JTVAC	CLPSO	DMS-PSO	HPSO-TVAC	FDR	PSO	HIDMS-PSO*
F1	6.9E-07	0.0E+00	8.0E+00	1.2E-13	4.7E-02	1.8E+02	1.0E-13	1.3E+03	1.4E+04	2.5E-09
F2	2.8E+03	3.8E-03	2.0E+03	9.6E+01	4.0E+04	1.9E+03	1.9E+02	1.1E+04	3.0E+04	2.8E+01
F3	1.1E+07	5.3E+03	1.4E+07	2.9E+06	1.5E+08	1.3E+07	4.4E+06	7.2E+07	1.6E+08	3.8E+06
F4	2.2E+04	2.2E+03	2.5E+04	2.7E+04	6.6E+04	1.0E+04	3.1E+04	2.6E+04	5.6E+04	2.5E+04
F5	7.5E+03	3.7E+03	6.3E+03	1.4E+04	1.5E+04	5.6E+03	1.6E+04	8.2E+03	1.4E+04	6.8E+03
F6	2.4E+02	5.0E+01	1.8E+05	1.3E+02	6.8E+03	1.9E+06	1.7E+02	9.9E+06	3.1E+09	1.2E+02
F7	6.2E+03	5.4E-01	6.2E+03	6.2E+03	6.2E+03	6.2E+03	6.2E+03	6.2E+03	8.1E+03	6.2E+03
F8	2.1E+01	2.1E+01	2.1E+01	2.1E+01	2.1E+01	2.1E+01	2.1E+01	2.1E+01	2.1E+01	2.1E+01
F9	1.1E+02	9.0E+01	1.8E+01	8.3E+01	1.2E+00	1.5E+02	1.1E+02	5.6E+02	1.9E+02	1.2E+02
F10	9.5E+01	1.0E+02	1.2E+02	1.6E+02	4.5E+02	1.5E+02	1.9E+02	4.3E+02	3.0E+02	1.3E+02

TABLE III
The mean error results obtained in CEC2017 test suite for problem size of 30 dimensions.

	BA	GWO	BOA	WOA	MFO	ABC	FPA	CS	IWO	PSO1	PSO2	DMS-PSO	HIDMS-PSO*
F1	7.3E+10	1.1E+09	3.0E+10	2.1E+06	8.1E+09	1.3E+02	1.1E+11	1.9E+04	3.0E+03	1.3E+11	1.3E+11	2.3E+03	4.7E+03
F3	2.2E+05	2.9E+04	6.7E+04	1.6E+05	7.7E+04	1.2E+05	1.8E+06	4.5E+04	6.4E+03	3.9E+08	3.9E+08	2.2E+03	2.3E-10
F4	2.1E+04	1.5E+02	2.5E+03	1.5E+02	5.1E+02	3.4E+01	3.6E+04	7.5E+01	8.8E+01	4.4E+04	4.4E+04	6.2E+01	6.1E+01
F5	5.1E+02	8.7E+01	3.3E+02	2.7E+02	1.8E+02	8.8E+01	6.2E+02	1.4E+02	4.1E+02	6.8E+02	6.8E+02	4.6E+01	5.2E+01
F6	1.1E+02	4.0E+00	6.4E+01	6.6E+01	2.5E+01	0.0E+00	1.3E+02	5.0E+01	7.2E+01	1.4E+02	1.4E+02	0.0E+00	0.0E+00
F7	1.5E+03	1.6E+02	5.1E+02	5.1E+02	3.5E+02	1.0E+02	2.5E+03	1.6E+02	2.0E+03	2.7E+03	2.7E+03	9.3E+01	8.7E+01
F8	4.3E+02	7.7E+01	2.9E+02	1.9E+02	1.7E+02	8.9E+01	5.6E+02	1.3E+02	3.5E+02	6.1E+02	6.1E+02	4.8E+01	4.8E+01
F9	2.1E+04	5.4E+02	6.9E+03	7.7E+03	5.1E+03	8.2E+02	3.1E+04	4.6E+03	7.6E+03	3.8E+04	3.8E+04	1.2E-01	2.6E+00
F10	8.8E+03	2.8E+03	7.7E+03	4.8E+03	4.1E+03	2.3E+03	9.1E+03	3.7E+03	4.7E+03	9.6E+03	9.6E+03	2.9E+03	2.7E+03

TABLE IV
The mean error results obtained in CEC2017 test suite for problem size of 50 dimensions.

	BA	GWO	BOA	WOA	MFO	ABC	FPA	CS	IWO	PSO1	PSO2	DMS-PSO	HIDMS-PSO*
F1	1.7E+11	4.6E+09	4.3E+10	7.1E+06	3.2E+10	9.2E+08	2.3E+11	1.4E+05	6.9E+03	1.3E+09	1.2E+10	3.6E+03	5.4E+03
F3	8.2E+07	7.0E+04	2.2E+05	7.8E+04	1.7E+05	6.6E+05	1.9E+08	1.6E+05	2.6E+04	9.6E+03	5.8E+04	3.0E+02	2.0E-04
F4	6.3E+04	4.3E+02	9.9E+03	2.8E+02	2.6E+03	1.2E+03	9.0E+04	7.7E+01	1.2E+02	2.5E+02	9.3E+02	1.4E+02	7.0E+01
F5	9.5E+02	1.7E+02	6.2E+02	4.2E+02	4.2E+02	5.0E+02	1.1E+03	2.9E+02	7.4E+02	2.3E+02	2.0E+02	1.3E+02	1.1E+02
F6	1.3E+02	1.1E+01	7.9E+01	7.6E+01	4.5E+01	3.0E+01	1.4E+02	6.2E+01	7.8E+01	2.0E+01	1.2E+01	5.3E-01	1.2E-01
F7	3.3E+03	3.0E+02	1.1E+03	9.9E+02	9.0E+02	5.7E+02	4.7E+03	3.4E+02	3.5E+03	2.8E+02	2.7E+02	2.0E+02	1.7E+02
F8	9.7E+02	2.0E+02	6.5E+02	4.1E+02	3.8E+02	5.0E+02	1.1E+03	2.8E+02	7.2E+02	2.3E+02	2.0E+02	1.3E+02	1.1E+02
F9	7.5E+04	3.7E+03	2.8E+04	1.9E+04	1.5E+04	3.0E+04	9.2E+04	1.6E+04	2.0E+04	5.8E+03	3.6E+03	2.1E+02	5.6E+01
F10	1.6E+04	5.6E+03	1.4E+04	9.1E+03	7.9E+03	1.5E+04	1.6E+04	7.0E+03	7.7E+03	6.5E+03	6.1E+03	5.1E+03	5.6E+03

TABLE V
Ranks of mean performance on CEC2005 test suite.

Algorithm	Avg(30D)	Final(30D)	Avg(50D)	Final(50D)
SRPSO	2.40	1	1.96	1
HCLDMS-PSO	3.52	2	4.32	4
HIDMS-PSO*	3.60	3	3.64	2
HCLPSO	4.32	4	4.24	3
DMS-PSO	5.16	5	4.68	5
HPSO-TVAC	5.52	6	5.96	7
MNHPSO-JTVAC	5.52	6	5.36	6
CLPSO	6.84	7	7.48	8
PSO	8.16	9	7.80	9
FDR-PSO	8.32	10	8.44	10

[5] Xue, Y., Tang, T., Pang, W. and Liu, A., 2020. Self-adaptive parameter and strategy based particle swarm optimization for large-scale feature selection problems with multiple classifiers. Applied Soft Computing, 88, p.106031.

[6] Sai Rayala, S. and Ashok Kumar, N., 2020. Particle Swarm Optimization for robot target tracking application. Materials Today: Proceedings,.

[7] M.R. Tanweer, S. Suresh, N. Sundararajan, Dynamic mentoring and self-regulation based particle swarm optimization algorithm for solving complex real-world optimization problems, Inform. Sci. 326 (2016) 1–24.

[8] Y. Shi, R. Eberhart, A modified particle swarm optimizer, in: Proceedings of the IEEE Congress on Evolutionary Computation, IEEE, 1998, pp. 69–73.

[9] M. Clerc, J. Kennedy, The particle swarm-explosion, stability, and convergence in a multidimensional complex space, IEEE Trans. Evol. Comput. 6 (1) (2002) 58–73.

TABLE VI
RANKS OF MEAN PERFORMANCE ON CEC2017 TEST SUITE.

Algorithm	Avg(30D)	Final(30D)	Avg(50D)	Final(50D)
HIDMS-PSO*	2.03	1	1.69	1
DMS-PSO	2.14	2	2.34	2
ABC	3.00	3	8.34	10
CS	3.79	4	4.03	3
GWO	5.17	5	5.00	4
MFO	6.45	6	8.10	9
IWO	6.76	7	6.93	7
WOA	7.17	8	7.79	8
BOA	8.00	9	9.69	11
BA	10.00	10	12.03	12
FPA	10.86	11	12.93	13
PSO^1	11.93	12	5.48	5
PSO^2	11.93	12	6.55	6

[10] A. Ratnaweera, S.K. Halgamuge, H.C. Watson, Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients, IEEE Trans. Evol. Comput. 8 (3) (2004) 240–255.

[11] M.S. Nobile, P. Cazzaniga, D. Besozzi, R. Colombo, G. Mauri, G. Pasi, Fuzzy Self-Tuning PSO: A settings-free algorithm for global optimization, Swarm Evol. Comput. 39 (2018) 70–85.

[12] R. Mendes, J. Kennedy, J. Neves, The fully informed particle swarm: simpler, maybe better, IEEE Trans. Evol. Comput. 8 (3) (2004) 204–210.

[13] J.-J. Liang, P.N. Suganthan, Dynamic multi-swarm particle swarm optimizer, in: Proceedings of the IEEE Swarm Intelligence Symposium, IEEE, 2005, pp. 124–129.

[14] J. Jie, W. Wang, C. Liu, B. Hou, Multi-swarm particle swarm optimization based on mixed search behavior, in: Proceedings of the 5th IEEE Conference on Industrial Electronics and Applications, 2010, pp. 605–610.

[15] W.H. Lim, N.A.M. Isa, Particle swarm optimization with increasing topology connectivity, Eng. Appl. Artif. Intell. 27 (2014) 80–102.

[16] J.-J. Liang, P.N. Suganthan, Dynamic multi-swarm particle swarm optimizer, in: Proceedings of the IEEE Swarm Intelligence Symposium, IEEE, 2005, pp. 124–129.

[17] Wang, S., Liu, G., Gao, M., Cao, S., Guo, A. and Wang, J., 2020. Heterogeneous comprehensive learning and dynamic multi-swarm particle swarm optimizer with two mutation operators. Information Sciences, 540, pp.175-201.

[18] X.-S. Yang, A new metaheuristic bat-inspired algorithm, Nature Inspired Cooperative Strategies for Optimization (NICSO 2010), Springer, 2010.

[19] S. Mirjalili, S.M. Mirjalili, A. Lewis, Grey Wolf Optimizer, Adv. Eng. Software 69 (2014) 46–61.

[20] S. Arora, S. Singh, Butterfly optimization algorithm: a novel approach for global optimization, Soft Comput. 23 (3) (2018) 715–734.

[21] S. Mirjalili, A. Lewis, The Whale Optimization Algorithm, Adv. Eng. Software 95 (2016) 51–67.

[22] S. Mirjalili, Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm, Knowl.-Based Syst. 89 (2015) 228–249. M.A. Awadallah.

[23] Yang, X. (2016). Nature-inspired optimization algorithms.: Elsevier, pp.256-258.

[24] D. Karaboga, B. Basturk, A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm, J. Global Optim. 39 (3) (2007) 459–471.

[25] N.H. Awad, M.Z. Ali, P.N. Suganthan, J.J. Liang, B.Y. Qu, Evaluation Criteria for the CEC 2017 Special Session and Competition on Single Objective Real-Parameter Numerical Optimization, 2016, Tech. Rep.

[26] A. Ratnaweera, S.K. Halgamuge, H.C. Watson, Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients, IEEE Trans. Evol. Comput. 8 (3) (2004) 240–255.

[27] Yang, X. (2016). Nature-inspired optimization algorithms.: Elsevier, pp.261-263.

[28] T. Peram, K. Veeramachaneni, C.K. Mohan, Fitness-distance-ratio based particle swarm optimization, in: Proceedings of the 2003 IEEE Swarm Intelligence Symposium, IEEE, 2003, pp. 174–181.

[29] A. Ratnaweera, S.K. Halgamuge, H.C. Watson, Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients, IEEE Trans. Evol. Comput. 8 (3) (2004) 240–255.

[30] Ghasemi, M., Akbari, E., Zand, M., Hadipour, M., Ghavidel, S. and Li, L., 2019. An Efficient Modified HPSO-TVAC-Based Dynamic Economic Dispatch of Generating Units. Electric Power Components and Systems, 47(19-20), pp.1826-1840.

[31] J.J. Liang, A.K. Qin, P.N. Suganthan, S. Baskar, Comprehensive learning particle swarm optimizer for global optimization of multimodal functions, IEEE Trans. Evol. Comput. 10 (3) (2006) 281–295.

[32] Tanweer, M., Suresh, S. and Sundararajan, N., 2015. Self regulating particle swarm optimization algorithm. Information Sciences, 294, pp.182-202.

[33] N. Lynn, P.N. Suganthan, Heterogeneous comprehensive learning particle swarm optimization with enhanced exploration and exploitation, Swarm Evol. Comput. 24 (2015) 11–24.

[34] Y. Shi, R. Eberhart, A modified particle swarm optimizer, in: Proceedings of the IEEE Congress on Evolutionary Computation, IEEE, 1998, pp. 69–73.

[35] Jain, M., Singh, V. and Rani, A. (2019). A novel nature-inspired algorithm for optimization: Squirrel search algorithm. Swarm and Evolutionary Computation, 44, pp.148-175.

[36] Kohli, M. and Arora, S. (2018). Chaotic grey wolf optimization algorithm for constrained optimization problems. Journal of Computational Design and Engineering, 5(4), pp.458-472.

[37] Mehrabian, A. and Lucas, C., 2006. A novel numerical optimization algorithm inspired from weed colonization. Ecological Informatics, 1(4), pp.355-366.

[38] Niu, B., Zhu, Y., He, X. and Wu, H., 2007. MCPSO: A multi-swarm cooperative particle swarm optimizer. Applied Mathematics and Computation, 185(2), pp.1050-1062.

[39] Tanweer, M., Auditya, R., Suresh, S., Sundararajan, N. and Srikanth, N., 2016. Directionally Driven Self-Regulating Particle Swarm Optimization algorithm. Swarm and Evolutionary Computation, 28, pp.98-116.