Solving high dimensional multimodal continuous optimisation problems using hybridisation between particle swarm optimisation variants

Hugo Deschenes* and Caroline Gagne

Department of Computer Science and Mathematics,
Université du Québec à Chicoutimi,
555, boulevard de l'Université,
Chicoutimi, QC, CA, G7H 2B1, Canada
Email: hugo.deschenes@uqac.ca
Email: caroline.gagne@uqac.ca
*Corresponding author

Abstract: This paper presents a comparison between three new hybridisations using three particle swarm optimisation (PSO) variants: The Barebones PSO (BPSO), the comprehensive learning PSO (CLPSO) and the cooperative learning PSO (CoLPSO). The goal of these hybridisations is to improve the exploration and the exploitation of the search space from these three variants and contributes to PSO on high scale continuous optimisation problems. The performance of these three new hybrids, named HCLBPSO-Half, HBPSO+CL and HCoCLPSO, are compared with the original methods on which they are based. The comparison is done using six classical continuous optimisation functions with dimensions set to 50, 100 and 200, and all 15 continuous optimisation functions from the CEC'15 benchmark with dimensions set to 10, 30, 50 and 100. The results are compared using the mean and median of executions.

Keywords: metaheuristics; continuous optimisation; particle swarm optimisation; PSO; hybridisation; variants; high dimensional problems.

Reference to this paper should be made as follows: Deschenes, H. and Gagne, C. (2020) 'Solving high dimensional multimodal continuous optimisation problems using hybridisation between particle swarm optimisation variants', *Int. J. Metaheuristics*, Vol. 7, No. 3, pp.239–264.

Biographical notes: Hugo Deschenes is currently pursuing his PhD at Université du Québec à Chicoutimi (UQAC), Quebec, Canada. His research interests include applications of the particle swarm optimisation algorithm on continuous and discrete optimisation problems and hybridisation techniques.

Caroline Gagne is a Professor in the Department of Computer Science and Mathematics at Université du Québec à Chicoutimi (UQAC), Quebec, Canada. She received her PhD in Operations Research at Université Laval, Quebec, Canada in 2000. Prior to that obtention, she also received an MBA from Université Laval and a Bachelors degree in Computer Science from UQAC. Her field of interests includes nature-inspired algorithms and multi-objective optimisation of job sequencing problems. She has several contributions in that field and many collaborations with Canadian and French industries.

This paper is a revised and expanded version of a paper entitled 'Hybridising particle swarm optimisation variants on continuous optimisation functions' presented at MIC/MAEB, Barcelona, 4–7 July 2017.

Copyright © 2020 Inderscience Enterprises Ltd.

1 Introduction

Metaheuristics are widely used methods for solving complex optimisation problems, of a continuous or combinatorial nature (Talbi, 2009). They are useful when an optimal solution is not mandatory to profit a shorter execution time. In the light of the 'no-free-lunch' theorem (Wolpert and Macready, 1997), there is no single method that effectively solves every known optimisation problem. This theorem states that if an algorithm works well in solving a particular problem, it will inevitably be less efficient over many others. However, one interesting idea to get closer to this idealistic situation is to combine the forces of several metaheuristics by putting each one's particular advantages up front. This technique is called hybridisation.

In operations research, hybridisation is a well-known area of study which has proven powerful and has recently accomplished a great deal (Blum and Raidl 2016). It can be made in many ways using two (or more) optimisation methods. It is possible to observe many examples of hybridisations in the literature, e.g., using an ant colony optimisation algorithm, a genetic algorithm, a variable neighbourhood search and a tabu search (Raidl 2006).

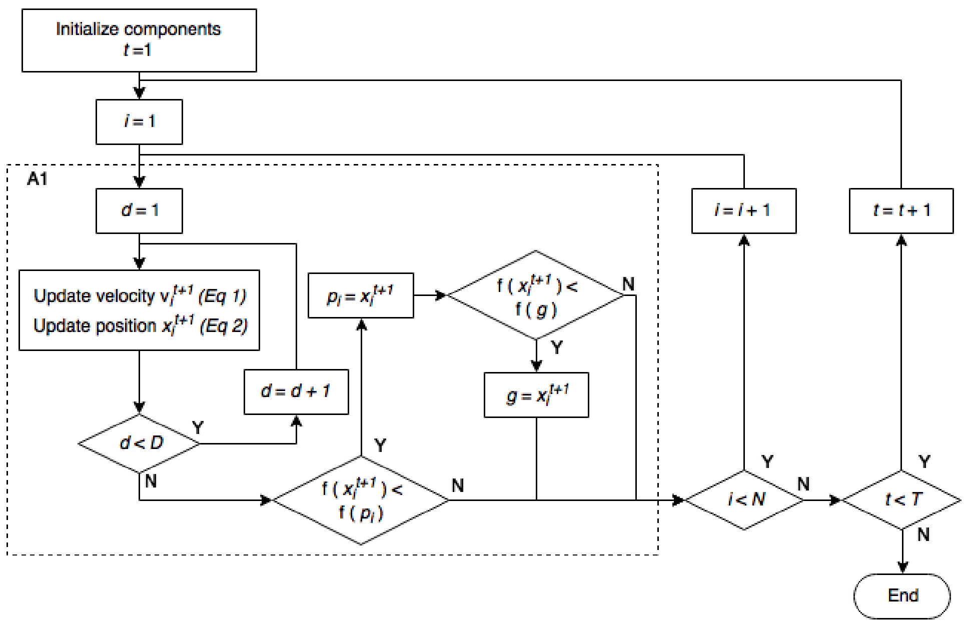
Banks et al. (2008), as well as Jordehi and Jasni (2015), highlighted many articles in which the particle swarm optimisation algorithm (PSO) is used in hybridisation. According to these authors, the PSO is a well-known metaheuristic that has been proven useful in solving continuous optimisation problems. Even if this method has been enhanced over the years to get better results, it has however demonstrated difficulties in solving high dimensional problems (Van den Bergh and Engelbrecht, 2004). Some of these enhancements change the nature of the PSO. Others add some components to its behaviour. Three variants have shown potential in solving various optimisation problems: The Barebones PSO (BPSO) (Kennedy, 2003), the comprehensive learning PSO (CLPSO) (Liang et al., 2006) and the cooperative learning PSO (CoLPSO) (Van den Bergh and Engelbrecht, 2004).

Because each of these three variants has performed well in solving continuous optimisation problems, it is natural to hybridise them to further improve the solutions and to contribute to PSO on large-scale continuous optimisation problems. The goal of this paper is to improve the PSO and its applicability in optimising high dimensional problems. It proposes three new hybrids based on the BPSO, the CLPSO and the CoLPSO which can get better results than these three methods executed separately. Section 2 reviews the literature on the PSO and its variants. Section 3 introduces three new PSO hybrids (HCLBPSO-Half, HBPSO+CL and HCoCLPSO). Section 4 shows the results of an analysis of these three new methods compared to the BPSO, the CLPSO and the CoLPSO. It also shows the results of the new hybrids compared to a recent algorithm, dynFWACM (Yu et al., 2015). This paper concludes in Section 5 by highlighting the best hybrid method and indicating some future work.

2 PSO evolution

The PSO (Eberhart and Kennedy, 1995) is an evolutionary algorithm where a population improves with the help of all individuals on each generation. It replicates the behaviour of birds in a flock or fish in a school as they are looking for food. It uses cooperation inside the swarm (flock, school) so that every particle (bird, fish) can improve with the aid of the rest of the swarm. When a particle is looking for a good solution (food source), it exchanges information with the swarm on the solutions observed so that all particles can potentially help each other to reach an optimum. The PSO algorithm evolved through the years. Variants have been proposed, like the BPSO, the CLPSO and the CoLPSO, to improve the results obtained with the PSO on different sets of problems.

Figure 1 Organisational chart for the PSO algorithm



Note: A1 frames the part of the algorithm responsible for the update of a particle

2.1 The PSO algorithm

Some features have been added to the original formulation of the PSO to help improve particle behaviour (Shi and Rc, 1998). This improvement is done by controlling the relative importance of each component used in the movement for each particle i in the swarm: its position (noted by x_i), its velocity (v_i), its best position so far (p_i), and the best global position of the swarm (g). A parameter (w) has been added for the inertial weight, which is used to moderate the velocity of a particle. Two coefficients, c_1 and c_2, are used to vary the importance of the best position visited by a particle and the best global position respectively. First, a particle i in generation t evolves in a swarm by updating its velocity v_i using equation (1), where the numbers r_1 and r_2 are randomly generated between 0 and 1. This particle is then updated using its position x_i (with dimension D) and the updated velocity according to equation (2).

$$v_i^{t+1} = w \cdot v_i^t + c_1 \cdot r_1 \cdot \left(p_i - x_i^t\right) + c_2 \cdot r_2 \cdot \left(g - x_i^t\right) \tag{1}$$

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

A summary of the PSO is illustrated in Figure 1. After initialising the components, the algorithm iterates on each particle i. $A1$ frames the part of the algorithm responsible for the update of a particle. It begins by updating its velocity and its position for each dimension. The particle is then evaluated according to the fitness function $f(\cdot)$ before being compared to its best-known position p_i and the best global position g of the swarm. The new value is copied at its place if the solution improved. The process goes on until all N particles have been updated for T generations.

A technique widely used in the literature is to vary the inertial weight (w) across the generations. This is done in order to help improve exploration at the beginning of the algorithm and exploitation at the end. Instead of using a fixed inertial weight, it is possible to reduce its value every generation. It has been shown that linearly lowering the inertial weight in each generation helps improve PSO performances (Eberhart and Shi 2001). According to these authors, its initial value (w_0) is set to 0.9 and its final value (w_1) is set to 0.4. This decrease is constant across the generations. A variant of this approach consists of using equation (3) as a formula to diminish the inertial weight value. This equation is based on the generation t and the maximum number of generations T (Ratnaweera et al., 2004). This technique is used for every new hybrid method that is explained in Section 3.

$$w = w_0 - \left(\frac{t\left(w_0 - w_1\right)}{T} \right) \tag{3}$$

2.2 Barebones PSO (BPSO)

Metaheuristics are well-known for using many parameters. This adds an additional challenge to optimisation. It is indeed necessary to find the best parameters to optimise algorithm performance. The BPSO helps optimisation by getting rid of parameters without compromising results (Kennedy, 2003), thus simplifying the challenge related to PSO.

The PSO swarm uses the best global position g and the best cognitive position p_i to update its velocity v_i. The position x_i obtained with these pieces of information acts like an oscillation centre for the other particles from the swarm as they are updated. When particles are getting close to each other, their velocities decrease significantly relative to the best cognitive position p_i and the best global position g. The author of the BPSO eliminated the concept of velocity from the PSO to prevent particles from moving directly towards potentially the same solution. He replaced the equation to update the position of a particle by an equation generating a number from a normal distribution among all the potential solutions. This update calculates a mean μ and a variance σ for each dimension d of a particle i using equation (4) and equation (5). The particle's position is then updated using equation (6) instead of equation (2) before being constricted within the scope.

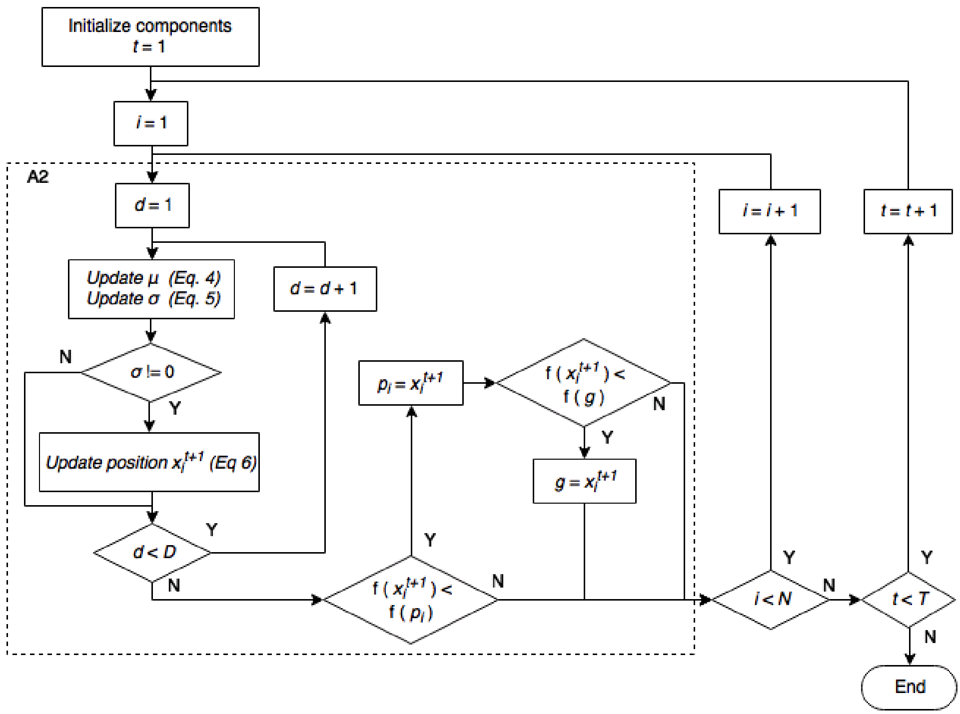
$$\mu = 0.5\left(p_i + g\right) \tag{4}$$

$$\sigma = \left|p_i - g\right| \tag{5}$$

$$x_i^{t+1} = \text{norm\_distr}(\mu, \sigma) \tag{6}$$

This modification allows the particles to prevent them from orbiting close to the same position for too many generations t. A summary of this method is illustrated in Figure 2. *A2* frames the part of the algorithm responsible for the update of a particle. It begins by calculating the mean and the variance before updating the new position. Once every dimension has been updated, the solution is evaluated before being compared with its best-known position p_i and the global best position g. The process is repeated until every particle is updated for every generation.

Figure 2 Organisational chart for the BPSO algorithm



Note: A2 frames the update of a particle responsible for the update of a particle.

Source: Kennedy (2003)

There are several variants of the BPSO in the literature. Whereas the BPSO uses a Gaussian distribution of numbers to update a particle's position, a variant uses a Lévy distribution instead (Richer and Blackwell, 2006). Other authors added a jump component to the displacement of a particle, thus allowing stuck solutions in local optima to jump elsewhere in the search space (Al-Rifaie and Blackwell, 2012). A similar process has also been elaborated with some minor differences, including the use of a Cauchy distribution of numbers (Krohling and Mendel, 2009).

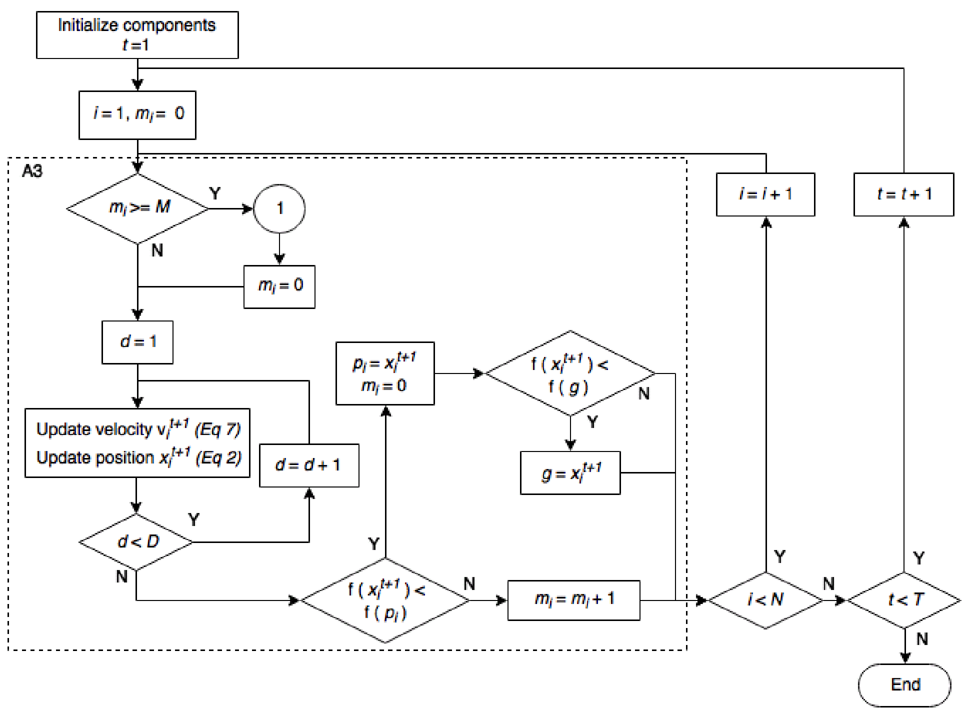
2.3 Comprehensive learning PSO

The CLPSO is a metaheuristic developed to help the particles to get out of a local optimum when optimising continuous multimodal functions (Liang et al., 2006). When the best global solution g stops evolving and stays the same during several generations, every particle is drawn toward this position even if it is not a good solution. CLPSO helps to palliate this problem by adding a comprehensive learning process modifying the best cognitive solution p_i and getting rid of the best global solution g in the update of a particle. Instead of using equation (1) from the PSO, the CLPSO uses the equation (7) to update the velocity of a particle.

$$v_i^{t+1} = w \cdot v_i^t + c_1 \cdot r_1 \cdot \left(p_i - x_i^t\right) \tag{7}$$

This algorithm is unique for its use of the comprehensive learning process, modifying the best cognitive solution p_i. Every time p_i is not improved when the update of a particle is completed, the value of the flag (m_i) corresponding to this particle is increased by 1. When it reaches the value M, chosen to be 7 by the authors, the comprehensive learning process is triggered. This modifies the composition of p_i based on a tournament selection for each dimension d of a particle i. Figure 3 presents a summary of this method, where N is the total number of particles in the swarm.

Figure 3 Organisational chart for the CLPSO



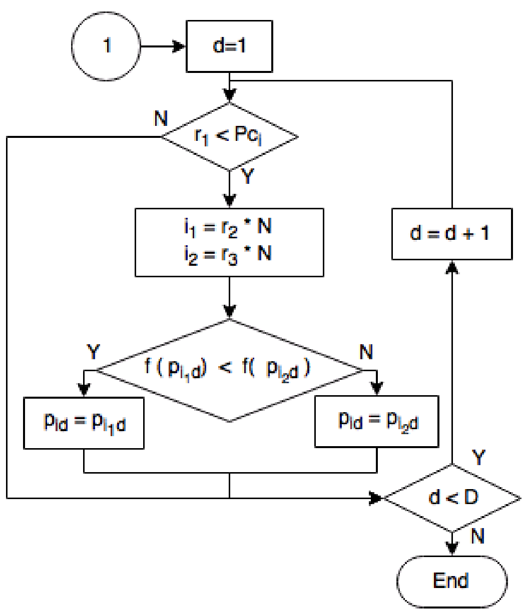
Note: A3 frames the part of the algorithm responsible for the update of a particle.
Source: Liang et al. (2006)

A3 frames the update of a particle. It begins by verifying if a particle has been stagnant for too many generations. If it has, the comprehensive learning process is triggered in ①, which is detailed in Figure 4. This process is based on a comprehensive probability Pc_i calculated specifically for each particle of the swarm. The value of this parameter is fixed according to equation (8), which is calculated at the beginning of the algorithm along with the other components. It remains constant for every generation.

$$Pc_i = 0.05 + 0.45 \cdot \frac{\left(\exp\left(\frac{10 \cdot (i-1)}{N-1} \right) - 1 \right)}{\left(\exp(10) - 1 \right)} \tag{8}$$

When the process is triggered as shown in Figure 4, a random number r_1 (between 0 and 1) is generated for each dimension d of a particle. If this number is lower than the comprehensive probability of the particle, then two other particles (i_1 and i_2) are randomly drawn from the swarm based on other random numbers (r_2 and r_3). Regarding their respective fitness score, the better of the two particles is used to copy the content of the dimension treated to the corresponding dimension in the particle *i*. $p_{i_1 d}$ and $p_{i_2 d}$ represent the value contained at the dimension d for the best cognitive solution p_i of the particle randomly selected (i_1 and i_2). Once it is completed, the algorithm pursue its course by updating the particle before evaluating it with the fitness function $f(\cdot)$. The best cognitive solution and the best global solution are then compared to the new solution that has been calculated with the particle. The update is repeated for T generations.

Figure 4 Organisational chart for the tournament selection for a particle *i* of the CLPSO



Source: Liang et al. (2006)

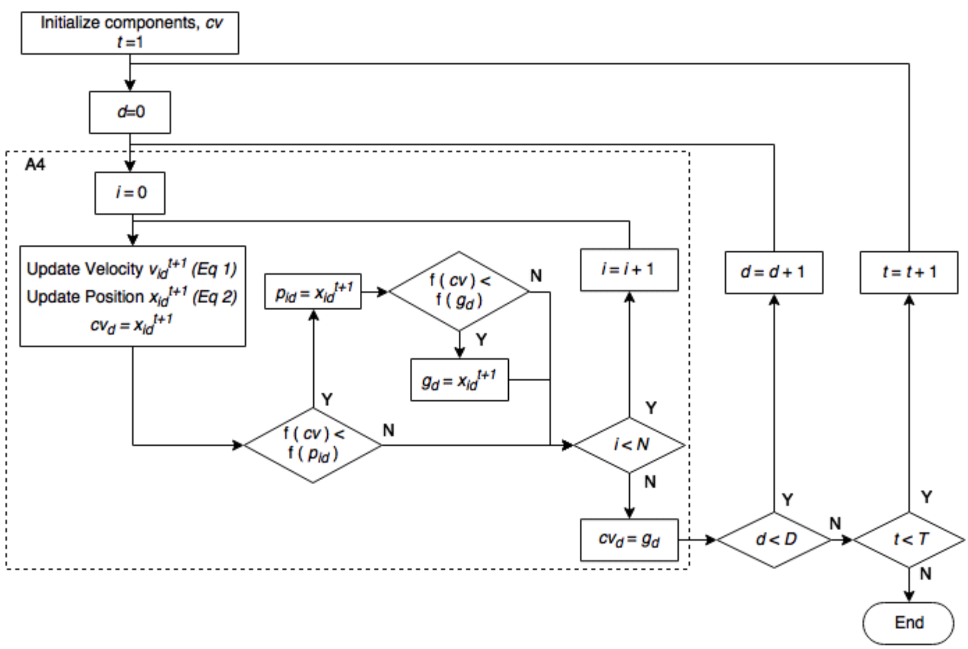
The CLPSO is used in the literature, among other problems, to solve the reactive power dispatch problem (Mahadevan and Kannan 2010) and the constrained mixed variable-optimisation problem (Gao and Hailu, 2010). Like the BPSO, the CLPSO has also been improved. Some authors added a concept of solution oppositions (Wang et al., 2011), which has previously proven to be useful in the differential evolution algorithm (Clerc, 1999).

2.4 Cooperative learning PSO (CoLPSO)

The CoLPSO is a metaheuristic developed to help optimise high scale continuous optimisation problems (Van den Bergh and Engelbrecht, 2004). The authors of this method indicate that the performance of the PSO declines the more dimension is added to the problem. This variant gets around this weakness by separately optimising each dimension of the standard PSO algorithm. To do so, the dimension loop and the particle loop from the PSO are swapped. Instead of updating all D dimensions for one particle at a time, the algorithm optimises N particles for each dimension individually. Instead of using only N particles (as it is in the standard PSO), this results in having $D*N$ particles where each dimension of the solution has its own population (sub-swarm) of N unidimensional particles. This allows the algorithm to update each dimension without being influenced by the fitness of the other dimensions from a single solution.

The use of a context vector (cv) is required to evaluate each particle. This is needed considering that each dimension d consists of a sub-swarm and that each of these sub-swarms cannot communicate between each other. The context vector contains the best solution obtained from each sub-swarms. Many swarm best solutions are set, one for each of these sub-swarms (g_d).

Figure 5 Organisational chart for the CoLPSO



Note: A4 frames the part of the algorithm responsible for the update of a sub-swarm.

When a sub-swarm is updated, the particles are evaluated by replacing their specific value in the corresponding dimension inside the context vector. The cv is then evaluated according to the fitness function $f(\cdot)$. This method is illustrated in Figure 5. A4 frames the part of the algorithm responsible for the update of a sub-swarm. It begins with the update of each particle i according to equation (1) and equation (2). The new generated position

x_{id}^{t+1} is then placed into the context vector at the corresponding dimension. The *cv* is evaluated according to the fitness function *f(·)*. If the new position helps to improve the *cv*, this position replaces the best-known solution p_{id} of the particle. The best global solution of the sub-swarm is also replaced the new position helped to improve the best global solution g_d of the sub-swarm. Once all *N* particles has been updated, the best value from the sub-swarm replaces the corresponding value in the *cv* if it helped enhance the fitness. This process is repeated for all *D* sub-swarms across *T* generations.

3 PSO hybridisations

A hybrid metaheuristic may have undergone modifications in several forms. Sometimes the nature of the optimisation method is preserved, other times some core processes are completely changed. As stated by Raidl (2015), choosing an adequate hybrid approach is determinant for achieving top performance in solving many real-world problems. Some authors suggested a classification method to better understand the complexity of this field of study, such as Talbi (2002) and Raidl (2015). They both give a taxonomy to classify hybrid metaheuristics.

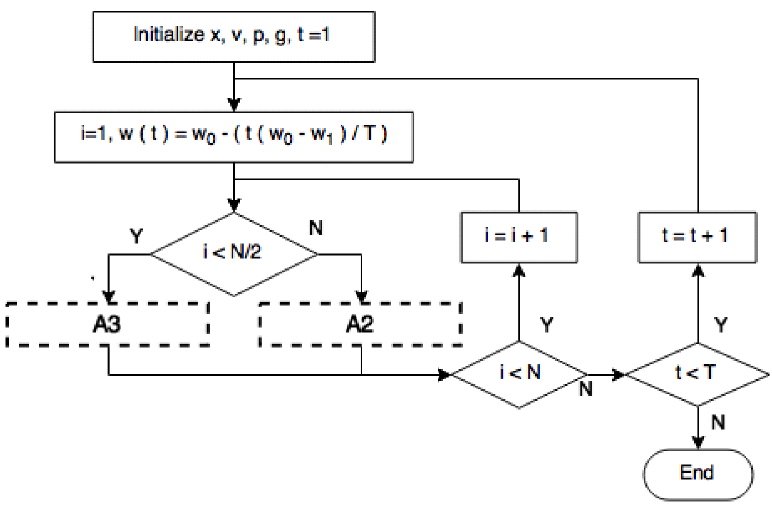
Talbi (2002) suggested a classification according to two criteria: the level (low-level/high-level) and the behaviour (relay/teamwork). A low-level indicates that a given function of a metaheuristic is replaced by another metaheuristic. A high-level indicates that metaheuristics that are used are self-contained and have no direct relationship with the internal workings of a metaheuristic. A relay hybrid indicates that a set of metaheuristics is applied one after the other as a pipeline. A teamwork hybrid indicates that the metaheuristics help each other in a parallel fashion. This classification is the one used in this paper for its simplicity.

Many PSO hybridisations exist in the literature. Each one of them tries to palliate the lacunas of the PSO by improving either the exploration or the exploitation of the search space (Van den Bergh and Engelbrecht, 2004). The methods described in this section propose three original hybridisations based on the BPSO, the CLPSO and the HCoCLPSO. The first hybrid divides the population to make half of the particles act like the BPSO and the other half act like the CLPSO. The second hybrid inserts the comprehensive learning process from the CLPSO at the beginning of the BPSO algorithm. The third hybrid uses the CoLPSO algorithm and adds the comprehensive learning process from the CLPSO at the beginning of the update of a sub-swarm. The main goal in these hybridisations is to ensure that most of the search space is surveyed, so that when the BPSO, the CLPSO or the CoLPSO can't explore some parts of it, one of the other algorithms compensates for it.

3.1 HCLBPSO-Half

The first new hybrid method is named HCLBPSO-Half (hybrid comprehensive learning barebones PSO). It is classified as a low-level teamwork algorithm according to Talbi (2002). This hybrid uses the benefits of both methods (BPSO and CLPSO) simultaneously, by separating the entire population into two groups of equal size: one with the behaviour of the BPSO, and the other with the behaviour of the CLPSO. The summary of this hybrid is illustrated in Figure 6. The assignment between the two selected methods is done at the beginning of the algorithm, right before setting the initial position of the population. At each generation, every particle will be updated regarding their respective method, beginning with the particles with CLPSO behaviour in Frame A3, which is the same algorithm as it is shown in Figure 3. The particles with the BPSO behaviour are then updated in Frame A2, which is the same algorithm as it is shown in Figure 2. The main advantage of this hybrid method is the mutual help between the groups of particles so that the exploration of the search space is improved. The BPSO uses the best global position *g* to determine the values of μ and σ. This best position can come either from a CLPSO particle or a BPSO particle. On one hand, this BPSO benefits from the exploration of the CLPSO. On the other hand, the CLPSO particles may trigger the comprehensive learning process after *M* stagnant generations of identical best cognitive solutions. During the tournament selection in this process, one of the two particles randomly selected comes from the BPSO population and the other from the CLPSO population. The CLPSO can then benefit from the exploration of the search space from the BPSO. Once the generation is completed, the best global position *g* is updated. With this hybrid, each method should benefit from the progression of the results obtained from the other one.

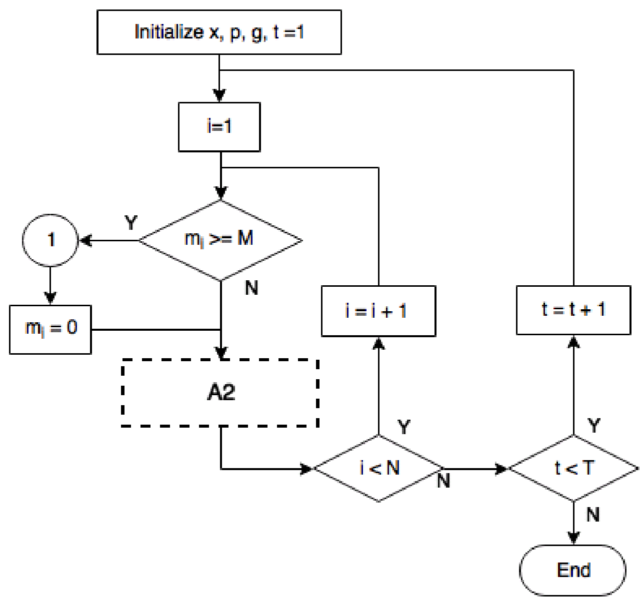
Figure 6 Organisational chart of HCLBPSO-Half with the particle division at the beginning of the algorithm



3.2 HBPSO+CL

The second new hybrid method is named HBPSO+CL (hybrid barebones PSO with comprehensive learning). It is classified as a high-level cooperative algorithm according to Talbi (2002). It adds the comprehensive learning process from the CLPSO at the beginning of the BPSO. The use of this component allows the BPSO to benefit from a more radical change in the composition of a solution when the fitness score for this particle is stagnant for several generations. Even if the Gaussian number generator from the BPSO helps the particles to get out of a local optimum, the use of the comprehensive learning process further increases this ability. This is done according to the comprehensive probability Pc_i of a particle *i* calculated with equation (8). This hybrid is illustrated in Figure 7. It begins with the activation of this comprehensive learning process if the flag m_i gets higher or equal to the value M. This process may change the fitness score of the best cognitive solution p_i. The algorithm continues with Frame A2 from the BPSO. This process is repeated for every particle and every generation t until all N particles have been updated and all T generations have been executed. These modifications are rather handy when many unidentical particles are in the same local optimum and quite close to each other.

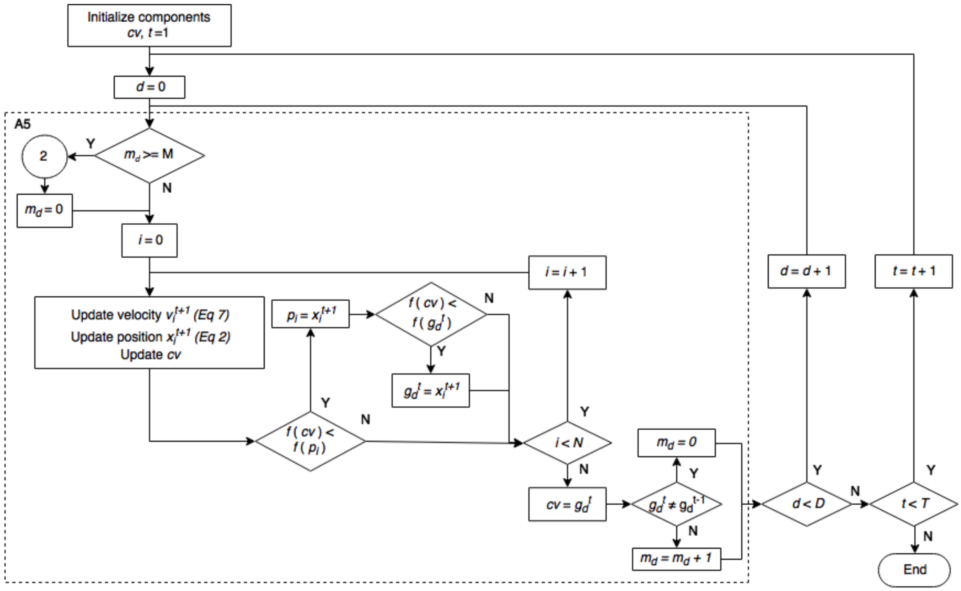
Figure 7 Organisational chart of HBPSO+CL



3.3 *HCoCLPSO*

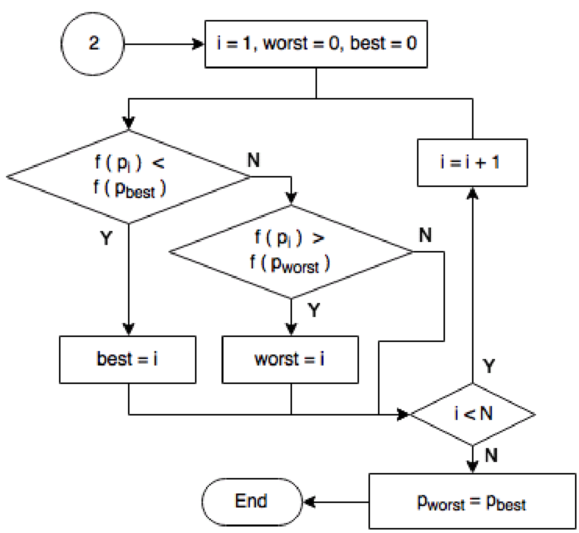
The third new hybrid method is named HCoCLPSO (hybrid cooperative comprehensive learning PSO). It is classified as a high-level cooperative algorithm according to Talbi (2002). It is inspired by the work of Maitra and Chatterjee (2008) that was used to solve the image segmentation problem using multilevel thresholding. Figure 8 illustrates this method where Frame A5 represents the part of the algorithm responsible for the update of a sub-swarm. This method adds a comprehensive learning process from the CLPSO into the CoLPSO structure for each sub-swarm. These sub-swarms from the CoLPSO might undergo changes due to stagnant generations according to the comprehensive learning process, thus helping to attain a good level of exploration of the search space. This process is triggered in ② if the flag m_d from each sub-swarm stays the same for at least M generations. As shown in Figure 9, instead of a tournament selection for the comprehensive learning process, the method finds the best and the worst particle for each dimension and replaces the worst particle with the best one. The rest of the algorithm in Figure 8 is similar to Frame A4 from the CoLPSO detailed in Section 2.4. The main difference can be observed with the addition of the flag m_d at the end of the update of a sub-swarm. The flag is increased by one if the sub-swarm's current best global solution g_d^t did not improve and is set to zero if it did improve.

Figure 8 Organisational chart of HCoCLPSO



Note: A5 frames the part of the algorithm responsible for the update of a sub-swarm.

Figure 9 Organisational chart of the process replacing the tournament selection from the CLPSO



4 Experimentation

Six classical continuous optimisation functions (Dixon and Szego, 1978; Molga and Smutnicki, 2005) and 15 continuous optimisation functions from the CEC'15 benchmark set (Liang et al., 2014) have been selected. The names and attributes of these functions are given in Table 1. They are divided into four categories: unimodal, multimodal, hybrid and composition. The chosen functions are significant in the way that they can be resolved with a dimension set to any number, thus being relevant in high-scale optimisation.

Table 1 List of all continuous optimisation functions used to compare algorithms

Attribute	*No.*	*Name of the function*
Classical		
Unimodal	A1	Sphere
	A2	Rosenbrock
Multimodal	A3	Ackley
	A4	Griewank
	A5	Rastrigin
	A6	Schwefel
CEC'15 benchmark set		
Unimodal	B1	Rotated high conditioned elliptic function
	B2	Rotated cigar function
Multimodal	B3	Shifted and rotated Ackley's function
	B4	Shifted and rotated Rastrigin's function
	B5	Shifted and rotated Schwefel's function
Hybrid	B6	Hybrid function 1 (3 functions)
	B7	Hybrid function 2 (4 functions)
	B8	Hybrid function 3 (5 functions)
Composition	B9	Composition function 1 (3 functions)
	B10	Composition function 2 (3 functions)
	B11	Composition function 3 (5 functions)
	B12	Composition function 4 (5 functions)
	B13	Composition function 5 (5 functions)
	B14	Composition function 6 (7 functions)
	B15	Composition function 7 (10 functions)

Each test function is configured to be solved in a search space of $[-100. 100]^D$. Experiments have been carried out for a dimension set to 50, 100 and 200 for each classical function, and a dimension set to 10, 30, 50 and 100 for the CEC'15 benchmark set. Most algorithms used a total number of particles (N) equal to 40, except for the CoLPSO and the HCoCLPSO. This number has been chosen according to Pluhacek et al. (2016). They suggested that PSO variants perform well using 40 particles. The CoLPSO and the HCoCLPSO both use a total number of particles equal to 20. The number is lowered compared to other algorithms because they need more evaluations on each iteration due to the need of a bigger population function properly. We conducted some empirical tests regarding the best value for these two algorithms and 20 particles is the one that allowed them to perform slightly better than using 40 particles. Coefficients c_1 and c_2 are both set to 1.49445, a number proven to be efficient for the PSO (Eberhart and

Shi, 2000). A varying inertial weight factor w is also used according to the diminution proposed by Ratnaweera et al. (2004), which is detailed in Section 2.1. w_0 is then set to 0.9, and w_1 to 0.4. This allows the algorithm to have a great balance between exploration and intensification of the search space. These parameters are set according to a study by Schutte and Groenwold (2005) stating that dynamic inertia reduction is less sensitive to parameter variations. They also proved that it performs better for problems of higher dimensionality.

The comparison between hybrids is done in a similar way as what is proposed in a recent paper by Yu et al. (2015). The total number of evaluations is set to *10,000 x D* for each method. Tests have been executed 50 times for each test function while retaining the history of the global best solutions among all executions. The mean and the median from the global best solutions are used to identify significant differences between hybrids and their original PSO variants. The best method among all six is then compared with the algorithm proposed by Yu et al. (2015). Note that for the CEC'15 benchmark set (Liang et al., 2014), a value of 100 was added to function 1, 200 to function 2, and so on for all 15 functions from the dataset. All these values were subtracted in further tables from the calculated values so that the best value remains 0 instead of the suggested increments. This has been done in order to better compare the results with the values provided by Yu et al. in their paper.

4.1 Results for classical functions

Tables 2 to 3 summarises the performance of the PSO variants (BPSO, CLPSO and CoLPSO) and all three hybrids (HBPSO+CL, HCLBPSO-Half and HCoCLPSO). These tables show the results on the 6 classical functions listed in Table 1. Table 2 shows the results for the PSO variants from which the hybrids are elaborated. Table 3 shows the results for the hybrids. The best and worst solutions, the medians, the mean values and the standard deviations are shown for each dimension tested. Note that all values less than $1\text{x}10^{-15}$ are rounded to 0 and considered equivalent.

Table 4 compares the median and mean values of each PSO variants with each hybrid. The numbers in bold and highlighted represent the best values for each type of data for all algorithms. The numbers in bold but not highlighted represent the second-best values. This table helps to determine which algorithm performs the best between all six algorithms.

Table 4 shows that the CoLPSO is the method that outperformed the others with a dimension set to 50 and 100. It obtained a higher number of best medians and mean values for these dimensions. The HCoCLPSO comes close second with pretty good results at these dimensions when compared to the other five algorithms. However, the HCoCLPSO is the method that shows better results when using a dimension set to 200 by having a higher number of best medians and mean values compared to all other algorithms.

Furthermore, the HCoCLPSO is getting more best values with an increasing dimension. On problems with a dimension set to 100, the HCoCLPSO begins to show promising results by obtaining six times second-best values and three times best values from medians and means. On problems with a dimension set to 200, it obtains two times second-best values and six times best values from medians and means. All best values are for multimodal functions. The better is the algorithm compared to others when more dimensions are added to the problem.

Table 2 Results for the BPSO, the CLPSO and the CoLPSO on 6 classical functions for a dimension set to 50, 100 and 200

		50D														
		BPSO					*CLPSO*					*CoLPSO*				
		Best	*Worst*	*Median*	*Mean*	*Std.*	*Best*	*Worst*	*Median*	*Mean*	*Std.*	*Best*	*Worst*	*Median*	*Mean*	*Std.*
Unimodal	A1	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.45E-01	7.27E-01	3.83E-01	3.95E-01	1.24E-01	6.31E-13	5.06E-10	9.34E-12	2.54E-11	7.23E-11
	A2	4.10E-01	1.01E+04	3.52E+01	3.21E+02	1.44E+03	4.05E+03	2.22E+04	1.03E+04	1.07E+04	3.97E+03	2.43E+00	2.09E+02	9.71E+01	9.79E+01	5.26E+01
Multimodal	A3	2.00E+01	2.00E+01	2.00E+01	2.00E+01	0.00E+00	2.00E+01	2.01E+01	2.00E+01	2.00E+01	1.20E-02	1.09E-06	8.17E-05	5.19E-06	1.06E-05	1.44E-05
	A4	0.00E+00	2.67E+00	3.70E-03	1.62E-01	6.11E-01	3.36E-03	3.63E-02	8.40E-03	1.01E-02	5.76E-03	4.58E-13	9.23E-02	5.90E-11	8.91E-03	1.84E-02
	A5	8.36E+01	3.01E+02	1.53E+02	1.68E+02	4.99E+01	1.50E+01	4.65E+01	2.56E+01	2.53E+01	6.52E+00	9.14E-09	5.45E-06	9.53E-08	3.88E-07	8.70E-07
	A6	1.80E+04	1.82E+04	1.81E+04	1.81E+04	4.71E+01	1.78E+04	1.80E+04	1.78E+04	1.78E+04	4.17E+01	1.78E+04	1.78E+04	1.78E+04	1.78E+04	1.10E-11
		100D														
		BPSO					*CLPSO*					*CoLPSO*				
		Best	*Worst*	*Median*	*Mean*	*Std.*	*Best*	*Worst*	*Median*	*Mean*	*Std.*	*Best*	*Worst*	*Median*	*Mean*	*Std.*
Unimodal	A1	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	4.82E-02	1.94E-01	1.07E-01	1.11E-01	3.05E-02	7.41E-12	6.02E-10	5.33E-11	9.23E-11	1.20E-10
	A2	1.46E-01	1.30E+03	1.51E+02	2.88E+02	3.61E+02	1.65E+03	1.00E+04	3.52E+03	3.97E+03	1.83E+03	7.18E+01	3.42E+02	1.79E+02	1.86E+02	6.62E+01
Multimodal	A3	2.00E+01	2.00E+01	2.00E+01	2.00E+01	0.00E+00	2.00E+01	2.01E+01	2.01E+01	2.01E+01	2.37E-02	6.07E-06	1.07E-03	2.85E-05	7.61E-05	1.63E-04
	A4	0.00E+00	7.74E+00	2.74E+00	2.67E+00	2.20E+00	6.28E-04	8.29E-03	1.18E-03	1.38E-03	1.06E-03	3.71E-12	5.62E-02	2.12E-09	6.29E-03	1.09E-02
	A5	2.53E+02	7.92E+02	4.34E+02	4.52E+02	1.22E+02	7.50E+01	1.60E+02	1.03E+02	1.05E+02	1.87E+01	6.26E-07	3.85E-03	6.57E-06	1.71E-04	6.43E-04
	A6	3.60E+04	3.64E+04	3.61E+04	3.61E+04	6.56E+01	3.56E+04	3.60E+04	3.58E+04	3.58E+04	8.31E+01	3.55E+04	3.55E+04	3.55E+04	3.55E+04	2.20E-11
		200D														
		BPSO					*CLPSO*					*CoLPSO*				
		Best	*Worst*	*Median*	*Mean*	*Std.*	*Best*	*Worst*	*Median*	*Mean*	*Std.*	*Best*	*Worst*	*Median*	*Mean*	*Std.*
Unimodal	A1	2.00E-32	8.88E-09	0.00E+00	1.78E-10	1.26E-09	1.72E-02	1.40E-01	4.02E-02	4.61E-02	2.35E-02	8.80E-11	2.03E-08	5.70E-10	1.39E-09	3.00E-09
	A2	4.10E-01	1.05E+04	3.53E+02	1.18E+03	2.77E+03	1.98E+03	1.16E+04	3.01E+03	3.70E+03	1.97E+03	2.29E+02	5.32E+02	3.70E+02	3.76E+02	7.48E+01
Multimodal	A3	2.00E+01	2.00E+01	2.00E+01	2.00E+01	0.00E+00	2.00E+01	2.02E+01	2.01E+01	2.01E+01	3.77E-02	3.19E-05	2.58E-02	1.90E-04	1.58E-03	4.81E-03
	A4	3.66E-15	7.55E+00	3.38E+00	3.24E+00	1.93E+00	7.69E-05	8.17E-03	2.12E-04	3.81E-04	1.13E-03	1.90E-11	3.19E-02	2.88E-10	3.54E-03	7.29E-03
	A5	4.63E+02	1.49E+03	7.50E+02	8.09E+02	2.59E+02	3.25E+02	6.31E+02	4.56E+02	4.61E+02	6.33E+01	4.56E-04	5.74E-01	7.70E-03	4.85E-02	1.11E-01
	A6	7.22E+04	7.25E+04	7.23E+04	7.23E+04	9.31E+01	7.20E+04	7.32E+04	7.24E+04	7.25E+04	2.33E+02	7.11E+04	7.11E+04	7.11E+04	7.11E+04	4.41E-11

Table 3 Results for the HBPSO+CL, the HCLBPSO-Half and the HCoCLPSO on 6 classical functions for a dimension set to 50, 100 and 200

		50D														
		BPSO					*CLPSO*					*CoLPSO*				
		Best	*Worst*	*Median*	*Mean*	*Std.*	*Best*	*Worst*	*Median*	*Mean*	*Std.*	*Best*	*Worst*	*Median*	*Mean*	*Std.*
Unimodal	A1	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.45E-01	7.27E-01	3.83E-01	3.95E-01	1.24E-01	6.31E-13	5.06E-10	9.34E-12	2.54E-11	7.23E-11
	A2	4.10E-01	1.01E+04	3.52E+01	3.21E+02	1.44E+03	4.05E+03	2.22E+04	1.03E+04	1.07E+04	3.97E+03	2.43E+00	2.09E+02	9.71E+01	9.79E+01	5.26E+01
Multimodal	A3	2.00E+01	2.00E+01	2.00E+01	2.00E+01	0.00E+00	2.00E+01	2.01E+01	2.00E+01	2.00E+01	1.20E-02	1.09E-06	8.17E-05	5.19E-06	1.06E-05	1.44E-05
	A4	0.00E+00	2.67E+00	3.70E-03	1.62E-01	6.11E-01	3.36E-03	3.63E-02	8.40E-03	1.01E-02	5.76E-03	4.58E-13	9.23E-02	5.90E-11	8.91E-03	1.84E-02
	A5	8.36E+01	3.01E+02	1.53E+02	1.68E+02	4.99E+01	1.50E+01	4.65E+01	2.56E+01	2.53E+01	6.52E+00	9.14E-09	5.45E-06	9.53E-08	3.88E-07	8.70E-07
	A6	1.80E+04	1.82E+04	1.81E+04	1.81E+04	4.71E+01	1.78E+04	1.80E+04	1.78E+04	1.78E+04	4.17E+01	1.78E+04	1.78E+04	1.78E+04	1.78E+04	1.10E-11
		100D														
		BPSO					*CLPSO*					*CoLPSO*				
		Best	*Worst*	*Median*	*Mean*	*Std.*	*Best*	*Worst*	*Median*	*Mean*	*Std.*	*Best*	*Worst*	*Median*	*Mean*	*Std.*
Unimodal	A1	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	4.82E-02	1.94E-01	1.07E-01	1.11E-01	3.05E-02	7.41E-12	6.02E-10	5.33E-11	9.23E-11	1.20E-10
	A2	1.46E-01	1.30E+03	1.51E+02	2.88E+02	3.61E+02	1.65E+03	1.00E+04	3.52E+03	3.97E+03	1.83E+03	7.18E+01	3.42E+02	1.79E+02	1.86E+02	6.62E+01
Multimodal	A3	2.00E+01	2.00E+01	2.00E+01	2.00E+01	0.00E+00	2.00E+01	2.01E+01	2.01E+01	2.01E+01	2.37E-02	6.07E-06	1.07E-03	2.85E-05	7.61E-05	1.63E-04
	A4	0.00E+00	7.74E+00	2.74E+00	2.67E+00	2.20E+00	6.28E-04	8.29E-03	1.18E-03	1.38E-03	1.06E-03	3.71E-12	5.62E-02	2.12E-09	6.29E-03	1.09E-02
	A5	2.53E+02	7.92E+02	4.34E+02	4.52E+02	1.22E+02	7.50E+01	1.60E+02	1.03E+02	1.05E+02	1.87E+01	6.26E-07	3.85E-03	6.57E-06	1.71E-04	6.43E-04
	A6	3.60E+04	3.64E+04	3.61E+04	3.61E+04	6.56E+01	3.56E+04	3.60E+04	3.58E+04	3.58E+04	8.31E+01	3.55E+04	3.55E+04	3.55E+04	3.55E+04	2.20E-11
		200D														
		BPSO					*CLPSO*					*CoLPSO*				
		Best	*Worst*	*Median*	*Mean*	*Std.*	*Best*	*Worst*	*Median*	*Mean*	*Std.*	*Best*	*Worst*	*Median*	*Mean*	*Std.*
Unimodal	A1	2.00E-32	8.88E-09	0.00E+00	1.78E-10	1.26E-09	1.72E-02	1.40E-01	4.02E-02	4.61E-02	2.35E-02	8.80E-11	2.03E-08	5.70E-10	1.39E-09	3.00E-09
	A2	4.10E-01	1.05E+04	3.53E+02	1.18E+03	2.77E+03	1.98E+03	1.16E+04	3.01E+03	3.70E+03	1.97E+03	2.29E+02	5.32E+02	3.70E+02	3.76E+02	7.48E+01
Multimodal	A3	2.00E+01	2.00E+01	2.00E+01	2.00E+01	0.00E+00	2.00E+01	2.02E+01	2.01E+01	2.01E+01	3.77E-02	3.19E-05	2.58E-02	1.90E-04	1.58E-03	4.81E-03
	A4	3.66E-15	7.55E+00	3.38E+00	3.24E+00	1.93E+00	7.69E-05	8.17E-03	2.12E-04	3.81E-04	1.13E-03	1.90E-11	3.19E-02	2.88E-10	3.54E-03	7.29E-03
	A5	4.63E+02	1.49E+03	7.50E+02	8.09E+02	2.59E+02	3.25E+02	6.31E+02	4.56E+02	4.61E+02	6.33E+01	4.56E-04	5.74E-01	7.70E-03	4.85E-02	1.11E-01
	A6	7.22E+04	7.25E+04	7.23E+04	7.23E+04	9.31E+01	7.20E+04	7.32E+04	7.24E+04	7.25E+04	2.33E+02	7.11E+04	7.11E+04	7.11E+04	7.11E+04	4.41E-11

Table 4 Comparison between the median and the mean values for all six methods with a dimension set to 50, 100 and 200 (see online version for colours)

		BPSO		*CLPSO*		*CoLPSO*		*HBPSO+CL*		*HCLBPSO-Half*		*HCoCLPSO*	
		Median	*Mean*	*Median*	*Mean*	*Median*	*Mean*	*Median*	*Mean*	*Median*	*Mean*	*Median*	*Mean*
50D													
Unimodal	A1	**0.00E+00**	**0.00E+00**	3.83E-01	3.95E-01	**9.34E-12**	**2.54E-11**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	2.92E-08	3.48E-08
	A2	3.52E+01	3.21E+02	**1.03E+04**	1.07E+04	9.71E+01	**9.79E+01**	**2.14E+01**	**3.88E+01**	7.07E+01	1.45E+02	9.67E+01	9.84E+01
Multimodal	A3	2.00E+01	2.00E+01	2.00E+01	2.00E+01	**5.19E-06**	**1.06E-05**	2.00E+01	2.00E+01	2.00E+01	2.00E+01	**7.61E-05**	**8.29E-05**
	A4	3.70E-03	1.62E-01	8.40E-03	1.01E-02	**5.90E-11**	**8.91E-03**	**0.00E+00**	1.57E-01	**0.00E+00**	1.45E-02	2.41E-08	**5.96E-03**
	A5	1.53E+02	1.68E+02	2.56E+01	2.53E+01	**9.53E-08**	**3.88E-07**	1.68E+02	1.73E+02	1.22E+02	1.25E+02	**1.33E-05**	**1.74E-05**
	A6	**1.81E+04**	**1.81E+04**	**1.78E+04**	**1.78E+04**	**1.78E+04**	**1.78E+04**	**1.81E+04**	**1.81E+04**	**1.81E+04**	**1.81E+04**	**1.78E+04**	**1.78E+04**
100D													
		BPSO		*CLPSO*		*CoLPSO*		*HBPSO+CL*		*HCLBPSO-Half*		*HCoCLPSO*	
		Median	*Mean*	*Median*	*Mean*	*Median*	*Mean*	*Median*	*Mean*	*Median*	*Mean*	*Median*	*Mean*
Unimodal	A1	**0.00E+00**	**0.00E+00**	1.07E-01	1.11E-01	**5.33E-11**	**9.23E-11**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	7.86E-08	9.22E-08
	A2	**1.51E+02**	2.88E+02	3.52E+03	3.97E+03	1.79E+02	**1.86E+02**	**1.31E+02**	2.50E+02	1.53E+02	2.04E+02	1.97E+02	**2.00E+02**
Multimodal	A3	2.00E+01	2.00E+01	2.01E+01	2.01E+01	**2.85E-05**	**7.61E-05**	2.00E+01	2.00E+01	2.00E+01	2.00E+01	**9.88E-05**	**1.05E-04**
	A4	2.74E+00	2.67E+00	1.18E-03	**1.38E-03**	**2.12E-09**	6.29E-03	2.48E+00	2.75E+00	0.00E+00	1.53E-02	**2.21E-08**	**4.83E-03**
	A5	4.34E+02	4.52E+02	1.03E+02	1.05E+02	**6.57E-06**	**1.71E-04**	4.45E+02	4.57E+02	4.20E+02	4.41E+02	**4.64E-05**	**5.35E-05**
	A6	3.61E+04	3.61E+04	**3.58E+04**	**3.58E+04**	**3.55E+04**	**3.55E+04**	3.62E+04	3.62E+04	3.62E+04	3.62E+04	**3.55E+04**	**3.55E+04**
200D													
		BPSO		*CLPSO*		*CoLPSO*		*HBPSO+CL*		*HCLBPSO-Half*		*HCoCLPSO*	
		Median	*Mean*	*Median*	*Mean*	*Median*	*Mean*	*Median*	*Mean*	*Median*	*Mean*	*Median*	*Mean*
Unimodal	A1	**0.00E+00**	**1.78E-10**	4.02E-02	4.61E-02	**5.70E-10**	1.39E-09	**0.00E+00**	5.83E-08	**0.00E+00**	**0.00E+00**	1.78E-07	2.07E-07
	A2	**3.53E+02**	1.18E+03	3.01E+03	3.70E+03	**3.70E+02**	**3.76E+02**	4.04E+02	1.71E+03	3.88E+02	8.89E+02	3.76E+02	**3.81E+02**
Multimodal	A3	2.00E+01	2.00E+01	2.01E+01	2.01E+01	**1.90E-04**	1.58E-03	2.00E+01	2.00E+01	2.00E+01	2.00E+01	**1.04E-04**	**1.09E-04**
	A4	3.38E+00	3.24E+00	2.12E-04	**3.81E-04**	**2.88E-10**	3.54E-03	3.40E+00	3.46E+00	**1.13E-14**	1.92E-01	1.01E-08	**1.68E-03**
	A5	7.50E+02	8.09E+02	4.56E+02	4.61E+02	**7.70E-03**	**4.85E-02**	7.45E+02	7.67E+02	1.34E+03	1.35E+03	**2.13E-04**	**2.11E-04**
	A6	**7.23E+04**	**7.23E+04**	7.24E+04	7.25E+04	**7.11E+04**	**7.11E+04**	**7.23E+04**	**7.23E+04**	7.24E+04	7.24E+04	**7.11E+04**	**7.11E+04**

Table 5 Comparison between the CLPSO, the CoLPSO and the HCoCLPSO with a dimension set to 200 (see online version for colours)

		200D														
		CLPSO					*CoLPSO*					*HCoCLPSO*				
		Best	*Worst*	*Median*	*Mean*	*Std.*	*Best*	*Worst*	*Median*	*Mean*	*Std.*	*Best*	*Worst*	*Median*	*Mean*	*Std.*
Unimodal	A1	1.72E-02	**1.40E-01**	4.02E-02	4.61E-02	2.35E-02	**8.80E-11**	2.03E-08	**5.70E-10**	**1.39E-09**	**3.00E-09**	1.08E-07	7.22E-07	1.78E-07	2.07E-07	1.03E-07
	A2	1.98E+03	**1.16E+04**	3.01E+03	3.70E+03	1.97E+03	2.29E+02	5.32E+02	**3.70E+02**	**3.76E+02**	7.48E+01	**1.84E+02**	5.67E+02	3.76E+02	3.81E+02	**7.40E+01**
Multimodal	A3	2.00E+01	**2.02E+01**	2.01E+01	2.01E+01	3.77E-02	**3.19E-05**	2.58E-02	1.90E-04	1.58E-03	4.81E-03	6.40E-05	2.19E-04	**1.04E-04**	**1.09E-04**	**2.65E-05**
	A4	7.69E-05	8.17E-03	2.12E-04	**3.81E-04**	**1.13E-03**	**1.90E-11**	**3.19E-02**	**2.88E-10**	3.54E-03	7.29E-03	3.34E-09	1.97E-02	1.01E-08	1.68E-03	4.18E-03
	A5	3.25E+02	**6.31E+02**	4.56E+02	4.61E+02	6.33E+01	4.56E-04	5.74E-01	7.70E-03	4.85E-02	1.11E-01	**8.67E-05**	5.57E-04	**2.13E-04**	**2.11E-04**	**8.46E-05**
	A6	7.20E+04	**7.32E+04**	7.24E+04	7.25E+04	2.33E+02	**7.11E+04**	7.11E+04	**7.11E+04**	**7.11E+04**	**4.41E-11**	**7.11E+04**	7.11E+04	**7.11E+04**	**7.11E+04**	**4.41E-11**

The results from CoLPSO and HCoCLPSO show that optimising each dimension separately helps to get good results on multimodal functions. The dominance of the HCoCLPSO over the CoLPSO on problems with a dimension set to 200 shows that the use of using a comprehensive learning process to get out of local optima is a good way to keep enhancing the results on high dimensional problems once they stop improving. To better prove the contribution from each method, Table 5 shows more detailed results for the test conducted with a dimension set to 200 with the HCoCLPSO and its two PSO variants from which it is elaborated: the CLPSO and the CoLPSO. The numbers in bold and highlighted represent the best values for each type of data on each function, except for the worst solutions for which the highlights represent the worst values instead. The highlighted numbers int Table 5 show that the HCoCLPSO outperforms its two PSO variants on multimodal functions by having more best median and mean values than the other algorithms. It may not have the highest number of best solutions, but it is the steadiest algorithm for it is the one with the highest number of best standard deviations. It is also the method that gets the least amount of worst solutions. In the light of these results, the HCoCLPSO is the method that shows great potential on high dimensional functions and that retains our attention to pursue further tests on the CEC'15 benchmark set.

4.2 Results for CEC'15 benchmark set

Tables 6 to 9 summarises the performance of the HCoCLPSO compared with the dynFWACM algorithm from Yu et al. (2015) on all 15 functions from the CEC'15 benchmark set for the competition on learning-based real-parameter single objective optimisation. Each dimension tested is shown separately in Table 6 (10D), Table 7 (30D), Table 8 (50D) and Table 9 (100D). The best and worst solutions, the medians, the mean values and the standard deviations are shown. A value is in bold and highlighted for each type of data in Tables 6 to 9 whenever a method has the best value over the other method, except for the worst solution for which it represents the worst value instead. The last line of each table, entitled total, shows the number of times a method gets the best value for each data type.

According to the results showed in Tables 6 to 9, dynFWACM is the best method for problems with a dimension set to 10, 30 and 50. It has more best solutions, medians, mean values and standard deviations and less worst solutions than HCoCLPSO. However, Table 9 shows that HCoCLPSO gets better results with a dimension set to 100. It has more best solutions, medians and mean values and less worst solutions than dynFWACM. It is also worth noticing that HCoCLPSO gets better results in multimodal functions, which corresponds to functions labelled as multimodal, hybrid and composition by the benchmark.

Tables 6 to 9 also shows that HCoCLPSO gets better results compared to dynFWACM with an increasing dimension value. The higher is the dimension, the better are the results. HCoCLPSO gets a significant decrease in the number of worst solutions, passing from 11 out of 15 worst solutions in Table 6 to 3 out of 15 worst solutions in Table 9. It also gets a significant increase in the number of best medians and mean values from a dimension set to 50 in Table 8 to a dimension set to 100 in Table 9. Experimentations would have been even more interesting if the CEC'15 benchmark proposed dataset with higher dimensionality, i.e., with a dimension set to 200 like it has been done with the classical functions.

Table 6 Comparison between the dynFWACM and the HCoCLPSO (see online version for colours)

		10D									
		dynFWACM					*HCoCLPSO*				
		Best	*Worst*	*Median*	*Mean*	*Std.*	*Best*	*Worst*	*Median*	*Mean*	*Std.*
Unimodal	B1	7.05E+03	**4.90E+05**	9.23E+04	1.11E+05	**9.88E+04**	**1.50E+03**	1.50E+03	**1.50E+03**	**1.50E+03**	5.77E+05
	B2	1.75E+01	3.30E+04	**6.26E+03**	**8.86E+03**	**8.90E+03**	**7.45E+00**	**3.36E+04**	6.33E+03	1.11E+04	1.10E+04
Multimodal	B3	**2.00E+01**	2.00E+01	**2.00E+01**	2.00E+01	**2.33E-04**	2.01E+00	2.00E+01	**2.00E+01**	**1.96E+01**	2.54E+00
	B4	6.96E+00	3.48E+01	**1.59E+01**	**1.67E+01**	**6.65E+00**	**5.97E+00**	**4.48E+01**	1.89E+01	2.07E+01	8.94E+00
	B5	1.30E+02	1.09E+03	**4.78E+02**	**5.18E+02**	2.40E+02	**1.19E+02**	**1.17E+03**	5.53E+02	5.92E+02	**2.13E+02**
Hybrid	B6	**3.59E+01**	7.31E+03	**8.30E+02**	**1.74E+03**	**1.97E+03**	3.06E+03	**5.57E+04**	8.54E+03	1.33E+04	1.20E+04
	B7	9.03E-01	**2.12E+00**	1.45E+00	1.45E+00	**2.31E-01**	**2.57E-01**	1.71E+00	**1.20E+00**	**1.22E+00**	3.29E-01
	B8	**1.23E+01**	8.03E+03	**1.24E+03**	**1.96E+03**	**2.13E+03**	1.12E+03	**1.15E+06**	4.15E+04	1.57E+05	2.57E+05
Composite	B9	**1.00E+02**	1.02E+02	**1.00E+02**	**1.00E+02**	**4.59E-01**	**1.00E+02**	**2.29E+02**	**1.00E+02**	1.22E+02	4.69E+01
	B10	**1.54E+02**	1.22E+03	**4.75E+02**	**5.47E+02**	**2.37E+02**	2.99E+02	**1.26E+05**	3.92E+03	1.10E+04	2.18E+04
	B11	**1.85E+00**	3.03E+02	**3.01E+02**	**1.85E+02**	1.47E+02	1.84E+02	**6.06E+02**	3.03E+02	3.42E+02	**9.32E+01**
	B12	1.08E+02	1.17E+02	1.13E+02	**1.13E+02**	**1.52E+00**	**1.02E+02**	**2.31E+02**	**1.04E+02**	1.15E+02	3.52E+01
	B13	**9.60E-02**	1.96E-01	**1.20E-01**	**1.21E-01**	**1.66E-02**	2.47E+01	**1.25E+03**	3.48E+01	5.82E+01	1.72E+02
	B14	**1.00E+02**	8.74E+03	**6.83E+03**	6.30E+03	**2.12E+03**	**1.00E+02**	**1.14E+04**	7.00E+03	**5.81E+03**	2.92E+03
	B15	**1.00E+02**	1.00E+02	**1.00E+02**	**1.00E+02**	1.44E-13	**1.00E+02**	1.00E+02	**1.00E+02**	**1.00E+02**	**0.00E+00**
Total		9	2	12	11	12	9	11	6	5	3

Source: Yu et al. (2015) with a dimension set to 10

Table 7 Comparison between the dynFWACM and the HCoCLPSO (see online version for colours)

		30D									
		dynFWACM					*HCoCLPSO*				
		Best	*Worst*	*Median*	*Mean*	*Std.*	*Best*	*Worst*	*Median*	*Mean*	*Std.*
Unimodal	B1	**2.55E+05**	1.34E+06	**5.57E+05**	**6.17E+05**	**3.49E+05**	6.60E+05	**6.37E+06**	3.02E+06	3.18E+06	1.43E+06
	B2	**2.93E+01**	1.46E+04	**2.15E+03**	**3.31E+03**	**3.59E+03**	3.64E+01	**1.50E+04**	3.44E+03	4.71E+03	4.19E+03
Multimodal	B3	**2.00E+01**	2.00E+01	**2.00E+01**	**2.00E+01**	**5.05E-06**	**2.00E+01**	**2.01E+01**	**2.00E+01**	**2.00E+01**	8.91E-03
	B4	7.26E+01	**2.55E+02**	1.21E+02	1.30E+02	3.80E+01	**4.68E+01**	1.89E+02	**1.00E+02**	**1.00E+02**	**2.98E+01**
	B5	1.96E+03	**4.88E+03**	3.32E+03	3.38E+03	6.98E+02	**1.55E+03**	4.21E+03	**2.82E+03**	**2.75E+03**	**5.25E+02**
Hybrid	B6	**2.91E+03**	8.35E+04	**2.06E+04**	**2.69E+04**	**1.90E+04**	1.03E+05	**3.23E+06**	6.55E+05	8.15E+05	7.30E+05
	B7	8.58E+00	2.00E+01	1.49E+01	1.46E+01	**2.57E+00**	**4.36E+00**	**8.04E+01**	**8.98E+00**	**1.24E+01**	1.44E+01
	B8	**3.86E+03**	5.00E+04	**2.25E+04**	**2.40E+04**	**1.32E+04**	4.33E+04	**1.53E+06**	3.99E+05	5.10E+05	3.79E+05
Composite	B9	1.06E+02	1.10E+02	1.08E+02	**1.08E+02**	**9.01E-01**	**1.04E+02**	**3.24E+02**	**1.05E+02**	1.55E+02	8.62E+01
	B10	**7.47E+03**	7.83E+04	**2.57E+04**	**3.15E+04**	**2.01E+04**	2.24E+05	**1.93E+06**	7.51E+05	8.48E+05	3.83E+05
	B11	**3.01E+02**	9.75E+02	**6.77E+02**	**6.72E+02**	**1.54E+02**	3.07E+02	**1.16E+03**	9.10E+02	8.71E+02	2.06E+02
	B12	1.10E+02	**2.02E+02**	1.15E+02	1.17E+02	**1.23E+01**	**1.06E+02**	1.11E+02	**1.08E+02**	**1.08E+02**	1.30E+00
	B13	**1.73E-02**	4.58E-02	**2.48E-02**	**2.62E-02**	**7.46E-03**	9.44E+01	**6.81E+03**	1.08E+02	2.43E+02	9.48E+02
	B14	4.15E+04	**4.61E+04**	4.52E+04	4.49E+04	**1.02E+03**	**3.12E+04**	3.70E+04	**3.39E+04**	**3.38E+04**	1.16E+03
	B15	**1.00E+02**	1.00E+02	**1.00E+02**	**1.00E+02**	1.75E-13	**1.00E+02**	1.00E+02	**1.00E+02**	**1.00E+02**	**0.00E+00**
Total		9	4	9	10	12	8	10	8	7	3

Source: Yu et al. (2015) with a dimension set to 30

Table 8 Comparison between the dynFWACM and the HCoCLPSO (see online version for colours)

		50D									
		dynFWACM					*HCoCLPSO*				
		Best	*Worst*	*Median*	*Mean*	*Std.*	*Best*	*Worst*	*Median*	*Mean*	*Std.*
Unimodal	B1	**8.16E+05**	**4.06E+06**	**1.67E+06**	**1.80E+06**	**7.14E+05**	1.37E+06	1.37E+06	6.44E+06	7.29E+06	4.63E+06
	B2	**7.00E-01**	**3.52E+04**	**1.30E+03**	**4.17E+03**	**8.01E+03**	9.09E+00	9.09E+00	7.68E+03	1.15E+04	1.20E+04
Multimodal	B3	**2.00E+01**	2.00E+01	**2.00E+01**	**2.00E+01**	**1.30E-06**	**2.00E+01**	2.00E+01	**2.00E+01**	**2.00E+01**	1.18E-02
	B4	1.27E+02	**4.13E+02**	2.33E+02	2.44E+02	5.88E+01	**1.15E+02**	1.15E+02	**1.89E+02**	**1.93E+02**	**4.41E+01**
	B5	4.54E+03	**7.58E+03**	5.64E+03	5.78E+03	7.00E+02	**3.21E+03**	3.21E+03	**4.69E+03**	**4.69E+03**	**6.26E+02**
Hybrid	B6	**2.83E+04**	1.90E+05	**8.33E+04**	**8.81E+04**	**3.82E+04**	2.75E+05	**2.75E+05**	1.73E+06	2.28E+06	1.40E+06
	B7	1.78E+01	**1.06E+02**	5.89E+01	**4.70E+01**	2.61E+01	**1.27E+01**	1.27E+01	**4.92E+01**	**4.70E+01**	**2.23E+01**
	B8	**1.30E+04**	1.53E+05	**6.09E+04**	**6.88E+04**	**4.00E+04**	6.35E+05	**6.35E+05**	2.23E+06	2.35E+06	1.15E+06
Composite	B9	**1.03E+02**	**4.79E+02**	**1.03E+02**	**1.11E+02**	**5.26E+01**	1.06E+02	1.06E+02	1.07E+02	1.53E+02	1.00E+02
	B10	**1.35E+04**	9.26E+04	**3.84E+04**	**4.21E+04**	**1.72E+04**	1.68E+05	**1.68E+05**	1.04E+06	1.03E+06	5.03E+05
	B11	3.06E+02	**1.60E+03**	**1.11E+03**	**1.08E+03**	2.44E+02	**3.03E+02**	3.03E+02	1.36E+03	1.32E+03	**2.25E+02**
	B12	1.21E+02	**2.05E+02**	2.03E+02	1.75E+02	3.80E+01	**1.07E+02**	1.07E+02	**1.10E+02**	**1.10E+02**	**1.12E+00**
	B13	**6.43E-02**	2.58E-01	**1.08E-01**	**1.17E-01**	**4.51E-02**	1.74E+02	**1.74E+02**	1.93E+02	1.93E+02	8.27E+00
	B14	5.29E+04	**8.39E+04**	**5.33E+04**	**5.49E+04**	**6.34E+03**	**4.95E+04**	4.95E+04	5.92E+04	5.97E+04	9.75E+03
	B15	**1.00E+02**	1.00E+02	**1.00E+02**	**1.00E+02**	2.78E-13	**1.00E+02**	1.00E+02	**1.00E+02**	**1.00E+02**	**0.00E+00**
Total		9	9	11	12	9	8	4	6	6	6

Source: Yu et al. (2015) with a dimension set to 50

Table 9 Comparison between the dynFWACM and the HCoCLPSO (see online version for colours)

		100D									
		dynFWACM					*HCoCLPSO*				
		Best	*Worst*	*Median*	*Mean*	*Std.*	*Best*	*Worst*	*Median*	*Mean*	*Std.*
Unimodal	B1	**1.95E+06**	**7.50E+06**	**3.92E+06**	**3.88E+06**	**1.24E+06**	6.88E+06	6.88E+06	1.02E+07	1.04E+07	2.34E+06
	B2	**1.30E+01**	**2.00E+04**	**1.16E+03**	3.66E+03	4.63E+03	1.45E+01	1.45E+01	1.67E+03	**3.08E+03**	**3.85E+03**
Multimodal	B3	**2.00E+01**	2.00E+01	**2.00E+01**	**2.00E+01**	**2.58E-06**	**2.00E+01**	2.00E+01	**2.00E+01**	**2.00E+01**	1.69E-02
	B4	4.55E+02	**9.29E+02**	6.09E+02	6.17E+02	1.07E+02	**4.29E+02**	4.29E+02	**5.59E+02**	**5.87E+02**	**8.98E+01**
	B5	1.02E+04	**1.65E+04**	1.28E+04	1.29E+04	1.23E+03	**9.19E+03**	9.19E+03	**1.23E+04**	**1.23E+04**	**1.11E+03**
Hybrid	B6	**1.90E+05**	1.43E+06	**4.43E+05**	**5.40E+05**	**2.78E+05**	1.82E+06	**1.82E+06**	4.38E+06	4.63E+06	1.83E+06
	B7	3.47E+01	**1.78E+02**	1.30E+02	1.33E+02	3.85E+01	**3.22E+01**	3.22E+01	**1.14E+02**	**1.15E+02**	**3.68E+01**
	B8	**7.72E+04**	4.32E+05	**1.63E+05**	**1.83E+05**	**7.67E+04**	5.41E+05	**5.41E+05**	2.88E+06	3.56E+06	2.03E+06
Composite	B9	1.11E+02	**1.15E+02**	1.13E+02	**1.13E+02**	**9.32E-01**	**1.09E+02**	1.09E+02	**1.11E+02**	2.86E+02	2.99E+02
	B10	1.35E+05	**4.72E+05**	2.43E+05	2.48E+05	**7.35E+04**	**3.29E+04**	3.29E+04	**2.00E+05**	**2.34E+05**	1.70E+05
	B11	**3.05E+02**	**2.90E+03**	**2.46E+03**	**2.27E+03**	6.86E+02	2.29E+03	2.29E+03	2.62E+03	2.64E+03	**1.73E+02**
	B12	1.20E+02	**2.05E+02**	1.26E+02	1.54E+02	3.84E+01	**1.07E+02**	1.15E+02	**1.18E+02**	**1.18E+02**	**1.02E+00**
	B13	**5.71E-01**	2.90E+00	**1.19E+00**	**1.36E+00**	**5.95E-01**	3.99E+02	**3.99E+02**	4.12E+02	4.12E+02	7.74E+00
	B14	1.24E+05	**1.66E+05**	1.40E+05	1.41E+05	**8.72E+03**	**1.09E+05**	1.09E+05	**1.09E+05**	**1.14E+05**	1.41E+04
	B15	1.02E+02	**1.11E+02**	1.04E+02	1.04E+02	**1.63E+00**	**1.00E+02**	1.00E+02	**1.03E+02**	**1.03E+02**	1.69E+00
Total		7	11	7	7	9	9	3	9	9	6

Source: Yu et al. (2015) with a dimension set to 100

5 Conclusions

In this research, we presented three new hybrid methods based on three PSO variants that performed well in the literature: the BPSO, the CLPSO and the CoLPSO. One of the hybrids, named HBPSO+CL, adds a specific component from the CLPSO at the beginning of the BPSO: the comprehensive learning process. A second hybrid, named HCLBPSO-Half, has half the population behaving like that of the BPSO, and the other half like that of the CLPSO. A third hybrid, named HCoCLPSO, uses the CoLPSO algorithm and adds a modified comprehensive learning process from the CLPSO at the beginning of the update of a sub-swarm. The first part of this paper was to determine if PSO hybrid methods can improve the results of the PSO variants from which they are conceived when solving high dimensional continuous optimisation problems. The original methods and the hybrids were tested and compared on six classical continuous optimisation functions from the literature: Sphere, Rosenbrock, Ackley, Griewank, Rastrigin, Schwefel. The second part of this paper was to compare the best hybrid with another algorithm from the literature on a more recent benchmark. The HCoCLPSO was then tested and compared with the dynFWACM on the CEC'15 benchmark set for the competition on learning-based real-parameter single objective optimisation. The goal was to bring out the best hybrid when solving multimodal high dimensional continuous optimisation problems.

The tests executed indicate that HCoCLPSO is the hybrid that offers overall better results than the other algorithms on high dimensional multimodal problems when solving classical functions and the CEC'15 benchmark set. The other two hybrids (HBPSO+CL and HCLBPSO-Half) performed well on low dimensional unimodal problems, but their performances lowered drastically with the use of higher dimensions.

With the HCoCLPSO showing the more promising results, this hybrid shows that it is possible to further improve the results while solving high dimensional multimodal problems with PSO. Hybridising the CoLPSO with another PSO variant, the CLPSO, helps to improve the results on such problems. It would be interesting in future works to test the HCoCLPSO on even higher dimensional problems and to hybridise the CoLPSO with other PSO variants to know which one improves the CoLPSO the most. It would also be interesting to test the HCoCLPSO with discretisation techniques to see how well it performs on combinatorial problems.

References

Al-Rifaie, M.M. and Blackwell, T. (2012) 'Bare bones particle swarms with jumps', *International Conference on Swarm Intelligence*, pp.49–60, Springer.

Banks, A., Vincent, J. and Anyakoha, C. (2008) 'A review of particle swarm optimization. Part Ii: hybridisation, combinatorial, multicriteria and constrained optimization, and indicative applications', Vol. 7, No. 1, pp.109–124.

Blum, C. and Raidl, G.R. (2016) *Hybrid Metaheuristics: Powerful Tools for Optimization*, Springer, Switzerland.

Clerc, M. (1999) 'The swarm and the queen: towards a deterministic and adaptive particle swarm optimization', *IEEE Congress on Evolutionary Computation*, IEEE.

Dixon, L.C.W. and Szego, G.P. (1978) 'The global optimization problem: an introduction', *Towards Global Optimization*, Vol. 2, pp.1–15.

Eberhart, R. and Kennedy, J. (1995) 'A new optimizer using particle swarm theory', *International Symposium on Micro Machine and Human Science*, pp.39–43.

Eberhart, R.C. and Shi, Y. (2000) 'Comparing inertia weights and constriction factors in particle swarm optimization', *IEEE Congress on Evolutionary Computation*, IEEE, pp.84–88.

Eberhart, R.C. and Shi, Y. (2001) 'Particle swarm optimization: developments, applications and resources', *IEEE Congress on Evolutionary Computation*, IEEE, pp.81–86.

Gao, L. and Hailu, A. (2010) 'Comprehensive learning particle swarm optimizer for constrained mixed-variable optimization problems', *International Journal of Computational Intelligence Systems*, Vol. 3, No. 6, pp.832–842.

Jordehi, A.R. and Jasni, J. (2015) 'Particle swarm optimisation for discrete optimisation problems: a review', *Artificial Intelligence Review*, Vol. 43, No. 2, pp.243–258.

Kennedy, J. (2003) 'Bare bones particle swarms', *IEEE Swarm Intelligence Symposium*, IEEE, pp.80–87.

Krohling, R. and Mendel, E. (2009) 'Bare bones particle swarm optimization with Gaussian or Cauchy jumps', *IEEE Congress on Evolutionary Computation*, IEEE, pp.3285–3291.

Liang, J., Qu, B., Suganthan, P. and Chen, Q. (2014) *Problem Definitions and Evaluation Criteria for the Cec 2015 Competition on Learning-Based Real-Parameter Single Objective Optimization*, Technical Report, Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou, China and Technical Report, Nanyang Technological University, Singapore.

Liang, J.J., Qin, A.K., Suganthan, P.N. and Baskar, S. (2006) 'Comprehensive learning particle swarm optimizer for global optimization of multimodal functions', *IEEE Transactions on Evolutionary Computation*, Vol. 10, No. 3, pp.281–295.

Mahadevan, K. and Kannan, P. (2010) 'Comprehensive learning particle swarm optimization for reactive power dispatch', *Applied Soft Computing*, Vol. 10, No. 2, pp.641–652.

Maitra, M. and Chatterjee, A. (2008) 'A hybrid cooperative–comprehensive learning based Pso algorithm for image segmentation using multilevel thresholding', *Expert Systems with Applications*, Vol. 34, No. 2, pp.1341–1350.

Molga, M and Smutnicki, C. (2005) 'Test functions for optimization needs', *Comput. Inform. Sci*., Vol. 101, pp.1–43.

Pluhacek, M., Kadavy, T., Senkerik, R., Viktorin, A. and Zelinka, I. (2016) 'Comparing selected Pso modifications on Cec 15 benchmark set', *IEEE Symposium Series on Computational Intelligence (SSCI)*, pp.1–6.

Raidl, G. (2006) 'A unified view on hybrid metaheuristics', in Almeida, F., Blesa Aguilera, M.J., Blum, C., Moreno Vega, J.M., Pérez Pérez, M., Roli, A. and Sampels, M. (Eds.): *Hybrid Metaheuristics*, Springer Berlin Heidelberg.

Raidl, G.R. (2015) 'Decomposition based hybrid metaheuristics', *European Journal of Operational Research*, Vol. 244, No. 1, pp.66–76.

Ratnaweera, A., Halgamuge, S. and Watson, H.C. (2004) 'Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients', *IEEE Transactions on Evolutionary Computation*, Vol. 8, No. 3, pp.240–255.

Richer, T.J. and Blackwell, T.M. (2006) 'The Lévy particle swarm', *IEEE International Conference on Evolutionary Computation*, IEEE, pp.808–815.

Schutte, J.F. and Groenwold, A.A. (2005) 'A study of global optimization using particle swarms', *Journal of Global Optimization*, Vol. 31, No. 1, pp.93–108.

Shi, Y. and Rc, E. (1998) 'A modified particle swarm optimizer', *IEEE International Conference on Evolutionary Computation*, pp.69–73.

Talbi, E-G. (2002) 'A taxonomy of hybrid metaheuristics', *Journal of Heuristics*, Vol. 8, No. 5, pp.541–564.

Talbi, E-G. (2009) *Metaheuristics: From Design to Implementation*, John Wiley & Sons, Inc., Hoboken, New Jersey, USA.

Van den Bergh, F. and Engelbrecht, A.P. (2004) 'A cooperative approach to particle swarm optimization', *IEEE Transactions on Evolutionary Computation*, Vol. 8, No. 3, pp.225–239.

Wang, W., Wang, H. and Rahnamayan, S. (2011) 'Improving comprehensive learning particle swarm optimiser using generalised opposition-based learning', *International Journal of Modelling, Identification and Control*, Vol. 14, No. 4, pp.310–316.

Wolpert, D.H. and Macready, W.G. (1997) 'No free lunch theorems for optimization', *IEEE Transactions on Evolutionary Computation*, Vol. 1, No. 1, pp.67–82.

Yu, C., Kelley, L.C. and Tan, Y. (2015) 'dynamic search fireworks algorithm with covariance mutation for solving the Cec 2015 learning based competition problems', *IEEE Congress on Evolutionary Computation*, IEEE, pp.1106–1112.