American Journal of Engineering Research (AJER)
e-ISSN: 2320-0847 p-ISSN : 2320-0936
Volume-6, Issue-4, pp-60-71
www.ajer.org
Research Paper Open Access

Primal-Dual Asynchronous Particle Swarm Optimisation (*pdAPSO*) Hybrid Metaheuristic Algorithm for Solving Global Optimisation Problems

Dada Emmanuel Gbenga[1], Joseph Stephen Bassi[1] and Okunade Oluwasogo Adekunle[2]

[1](Department of Computer Engineering, Faculty of Engineering, University of Maiduguri, Nigeria).
[2](Department of Computer Science, Faculty of Science, National Open University of Nigeria (NOUN), Abuja, Nigeria).

ABSTRACT: *Particle swarm optimization (PSO) is a metaheuristic optimization algorithm that has been used to solve complex optimization problems. The Interior Point Methods (IPMs) are now believed to be the most robust numerical optimization algorithms for solving large-scale nonlinear optimization problems. To overcome the shortcomings of PSO, we proposed the Primal-Dual Asynchronous Particle Swarm Optimization (pdAPSO) algorithm. The Primal Dual provides a better balance between exploration and exploitation, preventing the particles from experiencing premature convergence and been trapped in local minima easily and so producing better results. We compared the performance of pdAPSO with 9 states of the art PSO algorithms using 13 benchmark functions. Our proposed algorithm has very high mean dependability. Also, pdAPSO have a better convergence speed compared to the other 9 algorithms. For instance, on Rosenbrock function, the mean FEs of 8938, 6786, 10,080, 9607, 11,680, 9287, 23,940, 6269 and 6198 are required by PSO-LDIW, CLPSO, pPSA, PSOrank, OLPSO-G, ELPSO, APSO-VI, DNSPSO and MSLPSO respectively to get to the global optima. However, pdAPSO only use 2124 respectively which shows that pdAPSO have the fastest convergence speed. In summary, pdPSO and pdAPSO uses the lowest number of FEs to arrive at acceptable solutions for all the 13 benchmark functions.*

Keywords: *Asynchronous Particle Swarm Optimization (A-PSO), Swarm Robots, Interior Point Method, Primal-Dual, gbest, and lbest.*

I. INTRODUCTION

In Venayagomoorthy et al [1], the authors described PSO as is a stochastic population based algorithm that operates on the optimization of a candidate solution (or particle) centered to optimize a directed performance measure [2]. The first PSO algorithm was proposed by Kennedy and Eberhart [3] was based on the social behavior exemplified by a flock of bird, a school of fish, and herds of animals. The algorithm uses a set of candidates called particles that undergo gradual changes through collaboration and contest among the particles from one generation to the other. PSO have been used to solve non-differentiable [4], non-linear [5], and non-convex engineering problems [6].

Abraham, Konar and Das [7] opined that PSO is theoretically straightforward and does not require any sophisticated computation. PSO uses a few parameters, which have minimal influence on the results unlike any other optimization algorithms. This property also applies to the initial generation of the algorithm. The randomness of the initial generation will not affect the output produced. Despite these advantages, PSO faces similar shortcomings as other optimization algorithms. Specifically, PSO algorithm suffers from premature convergence, inability to solve dynamic optimization problems, the tendency of particles to be trapped in the local minima and partial optimism (i.e., which degrades the regulation of its speed and direction).

Hypothetically, particle swarm optimization (PSO) is an appropriate tool for addressing many optimisation problem in different fields of human endeavour. Notwithstanding the optimal convergence rate of PSO, the algorithm is not able to efficiently handle dynamic optimization jobs which is very crucial in some fields such as stock market prediction, crude oil price prediction, swarm robotics and space project. This explains the unfavourable inadequacies that led to the recent drifts of development new variants of PSO algorithms for solving complex tasks. It has become the custom to develop hybrid PSO heuristic algorithms to tackle the weaknesses of some existing variants of PSO. The idea behind the development of *pdAPSO* is to improve the performance of PSO in solving global optimization problems through the hybridization of APSO and Primal Dual algorithms. In this study, we proposed a fusion of Asynchronous PSO with Primal-Dual Interior-Point method to resolve those common issues relevant to PSO algorithm. Two key components of this implementation are the explorative capacity of PSO, and the exploitative capability of the Primal-Dual Interior-Point algorithm. On the one hand, exploration is key in searching (i.e., traversing the search landscape) to provide reliable approximation values of the global optimal [8]. On the other, exploitation is critical to focusing the search on the ideal solutions resulting from exploration to produce more refined results [9].

The state-of-the-art Interior-Point algorithm has gained popularity as the most preferred approach for providing solution to large-scale linear programming problems [10]. They are however limited due to their inability to solve problems that are unstable in nature. This is because contemporary Interior-Point algorithms are not able to cope with the increasing need of the large number of constraints. Efforts to increase the efficiency of the Interior-Point algorithm have led to the development of another variant of this algorithm that can handle unstable linear programming problems. These algorithms lower the number of work per iteration by using small number of constraints thereby reducing the computational processing time drastically [11].

II. REVIEW OF RELEVANT WORK

New variants of the PSO algorithm can be developed by fusing it with an already tested approaches which have been successfully applied to solve complex optimization problems. Academicians and researchers have improved the performance of PSO by integrating into it the basics of other famous methods. Some researchers have also made efforts to increase the performance of popular evolutionary algorithms like Genetic Algorithm, Ant Colony and Differential Evolution, etc. by infusing the position and velocity update equations of the PSO. The purpose of the integration is to make PSO overcome some of its setbacks like premature convergence, particles been trapped in the local minima, and partial optimization.

Robinson et al. [12] developed the GA-PSO and PSO-GA and used them to solve a specific electromagnetic application problem of projection antenna. The results of their experiments revealed that the PSO-GA hybrid algorithm performs better than the GA-PSO, standard PSO only and GA only. He proposed the hybridization of GA and Hill Climbing algorithm the same year and used it to solve unconstrained global optimization problems [13]. Conradie, Miikkulainen, and Aldrich [14], developed the symbiotic neuro memetic evolution (SMNE) algorithm when they hybridized PSO and ‘symbiotic genetic algorithm’ and used it for neural network control devices in a corroboration learning context. Grimaldi et al. [15] developed the genetical swarm optimization (GSO) by hybridizing PSO and GA. They later went ahead and used their algorithm to solve combinatorial optimization problems. The presented different hybridization approaches [16]. They authenticated the genuineness of GSO using different multimodal benchmark problems and applied it in different domain as demonstrated by the authors in Gandelli et al. [17], Grimaccia et al. [18] and Gandelli et al. [19]. Hendtlass [20] proposed the Swarm Differential Evolution Algorithm (SDEA) where PSO swarm acts as the population for Differential Evolution (DE) algorithm, and the DE is carried out over some generations. After the DE have performed its part in the optimization, the resulting population is then optimized by PSO. Talbi and Batauche [21] developed the DEPSO algorithm and used it to solve problem in the field of medical image processing. In Hao et al. [22] introduced another variant of DEPSO where some probability distribution rules are used any of PSO or DE to produce the best solution. Omran et al. [23] developed a Bare Bones Differential Evolution (BBDE) algorithm which used the idea of barebones PSO and self-adaptive DE approaches. They used their algorithm to solve image categorization problem. Jose et al. [24] developed another variant of DEPSO algorithm that uses the differential modification systems of DE to update the velocities of particles in the swarm. Zhang et al. [25] proposed the DE-PSO algorithm that uses three unconventional updating approaches. Liu et al. [26] developed the PSO-DE algorithm that combines DE with PSO and uses the DE to update the former best positions of PSO particles to make them escape local magnetizers thereby avoiding inertia in the population. Capanio et al. [27] proposed a Superfit Memetic Differential Evolution (SFMDE) algorithm which is a hybrid of DE, PSO, Nelder Mead algorithm and Rosenbrock algorithm. The algorithm was used to solve some standard benchmark and engineering problems.

The researchers in Xu and Gu [28] developed the particle swarm optimization with prior crossover differential evolution (PSOPDE). Pant et al. [29] reported a DE-PSO algorithm that uses DE for the initial optimization process and then moved on to the PSO segment if DE fails to satisfy the optimum conditions. Khamsawang et al. [30] introduced another hybrid algorithm name PSO-DE that centres on standard PSO and DE. They used their algorithm to solve economic dispatch (ED) problem having constraints. Shelokar et al. [31] developed PSO with Ant Colony Optimization (PSACO) algorithm. The algorithm has two phases. The PSO is employed in the first phase and the result of the optimization is feed into ACO for the second phase of the optimization. In Hendtlass and Randall [32], ACO was integrated into PSO by Hendtlass and Randall. The best position is selected from the list of best positions obtained and recorded. Victoire and Jeyakumar [33] proposed the hybrid of PSO and sequential quadratic programming (SQP). It was used to solve economic dispatch problem in Boggs and Tolle [34]. Grosan et al. [35] developed an independent neighborhoods particle swarm optimization (INPSO) algorithm that is made up of autonomous sub-swarms that allows the production of many points at the end of iteration.

In Liu et al. [36], the authors developed a turbulent PSO (TPSO) in the effort to surmount the shortcomings of the traditional PSO. They later integrate TPSO with a fuzzy logic controller to make a Fuzzy Adaptive TPSO (FATPSO). Sha and Hsu [37] proposed a novel hybrid algorithm that combine PSO with Tabu search (TS) and applied it to solve job shop problem (JSP). He and Wang [38] developed a hybrid algorithm that fuse PSO and Simulated Annealing (SA) together. Mo et al. [39] introduced Particle Swarm Assisted Incremental Evolution Strategy (PIES). This algorithm uses PSO for global optimization while Evolutionary strategy is used for local optimization. Fan and Zahara [40] and Fan et al. [41] proposed the NM-PSO algorithm by integrating PSO with Nelder Mead Simplex method. Their work was later extended by Zahara and Kao [42] and used to solve constricted optimization tasks. Guo et al. [43] proposed an algorithm that hybridized PSO with gradient descent (GD) method and they used it for fault identification. Shen et al. [44] introduced the HPSOTS algorithm which is a hybrid of PSO and Tabu search. Ge et al. [45] developed a hybrid of PSO and Artificial Immune System (AIS). Song et al. [46] proposed hybrid particle swarm cooperative optimization (HPSCO) algorithm merging simulated annealing algorithm and simplex method.

The research work of Kao et al. [47] presented an algorithm that combine NM-PSO algorithm developed in [40, 41], with K-means algorithm and used for data clustering. Murthy et al. in [48] proposed an algorithm that have the advantages of the parameter-free PSO (pf-PSO) and the extrapolated particle swarm optimization like algorithm (ePSO). Kuo et al. in [49] proposed the HPSO algorithm that amalgamated a random-key(RK) encoding system, individual enhancement (IE) system, and particle swarm optimization (PSO) and used to solve the flow-shop scheduling tasks. Chen et al. [50] developed the PSO-EO algorithm by hybridizing of PSO with Extremal Optimization (EO) as reported in Boettcher and Percus [51]. Kavehand and Talatahari [52, 53] proposed a heuristic particle swarm ant colony optimization (HPSACO) and a discrete heuristic particle swarm ant colony optimization (DHPSACO). Wei et al. [54] introduced the concept of entrenching swarm targets into Fast Evolutionary Programming (FEP) algorithm to make the swarm’s performance better. Pant et al. in [55] presented an AMPSO algorithm which combines PSO and EP mutation operator employing Beta distribution.

The (VL-ALPSO) was proposed by Tang and Eberhard in [56] to make planning for change in the physical position of swarm robots for collective search of targets more effective. The authors explained that the VL-ALPSO approach to swarm robotics is the amalgamation of augmented Lagrangian multipliers, velocity restrictions in addition to virtual detectors to guarantee the implementation of constraints, obstacle avoidance and mutual avoidance which are situations obtainable in swarm mobile robots in coordinated movements. Augmented Lagrangian Particle Swarm Optimization (ALPSO) algorithm was presented by Sedlaczek and Eberhard in [57]. The authors made use of some part of the original PSO technique and combines it with Augmented Lagrangian Multiplier.

III. PARTICLE SWARM OPTIMIZATION (PSO) ALGORITHM

PSO was originally proposed by James Kennedy and Russell Eberhart in 1995 as seen in [58]. The algorithm is made up of particles which have position and velocity. Each of the particles of a swarm epitomizes a possible solution in PSO. Each of the particles of a swarm epitomizes a possible solution in PSO. The particles explore the problem search space seeking for the best or at least a solution that is suitable. Each of the particles changes their movement according to their own accumulated knowledge of moving in the environment and that of their neighbours. In PSO (X_i) represent the position of a particle, and (V_i) the velocity of the particle. The particle’s number is i. Where (i = 1,…,N), and N is the number of particles in the swarm. The i^{th} particle is denoted as X_i. The velocity is the degree at which the subsequent position is varying as regards the present position. V_i represent the velocity for the particle i. As the algorithm begins, the position and velocity of the particles are given numerical values haphazardly. This is followed by using equations (1) and (2) to update the position and velocity of the particles after successive iterations are conducted throughout the search.

$$v_{i,m}^{(t+1)} = w * v_{i,m}^{(t)} + c_1 * rand_1() * (pbest_{i,m} - x_{i,m}^{(t)}) + c_2 * rand_2() * (gbest_m - x_{i,m}^{(t)}) \quad (1)$$

$$x_{i,m}^{(t+1)} = x_{i,m}^{(t)} + v_{i,m}^{(t+1)} \quad (2)$$

As opined by Shi and Eberhart in their statement in [5], to prevent commotion, the value $v_{i,m}^{(t+1)}$ is fixed at $\pm v_{max}$. The reason is that the value of v_{max} is going to be extremely large if the scope of search is too broad. Also, if v_{max} is very narrow, the extent of the search will be unreasonably reduced thereby forcing the particles to do local exploration. The inertia weight is represented as *w* (constriction factor) is the inertia parameter; this regulates algorithm’s searching properties. Shi and Eberhart posited as seen in [5] that it is better to commence the search using a larger inertia value (a more global search) that is automatically decreased to the end of the optimization (a more local search). Using inertia weight with smaller values mostly ensures fast convergence as little time is wasted on the exploration of the global space [59]. The inclusion of *w* in the equation is to provide equilibrium between the global and local search capability of the particles. There are two techniques that have been presented for the choice of suitable values for inertia factor. The number one technique is called linear method, here the inertia weight decreases linearly after each iteration until the highest number of iteration or the highest number of inertia parameter is reached [60].

$$w_{i+1} = w_{max} - \frac{w_{max} - w_{min}}{i_{max}} i$$

The number two technique is called the dynamic method, here the value of the inertia reduces from the initial value to the final value fractionally by Δ_w.

$$w_{i+1} = \Delta_w w_i$$

Where the value of Δ_i varies from 1 to 0. Judging from the results of experiments that have been performed, the performance of the dynamic method in term of convergence is superior to the one of linear method. In [5], it was showed that PSO having different swarm population has practically alike but not identical performance. The c_1 and c_2 are two positive constants representing the cognitive scaling and social scaling factors which according to [60] are usually set to 2. The stochastic variable $rand_1()$ and $rand_2()$ has the distribution U (0, 1). These random variables are stand-alone functions that infuse momentum to the particles. The most ideal position located so far by the particle is denoted as $pbest_{i,m}$. The best position attained by the neighbouring particles is denoted as $gbest_m$. There are two types of particles neighbourhood in PSO, and the type of neighbourhood is what determines the value of $gbest_m$. The two types of neighbourhood are:

1. *gBest* (Global neighbourhood) – Here, there is a full connection among the particles, and the exploration of swarm is controlled by the best particle in the swarm.
2. *lBest* (Local neighbourhood) – There is no full connection among the particles in the swarm, rather they are connected only to their neighbours.

Equation 2 is used in updating the position of the particles whereby the velocity is added together with the earlier position and a new search is started from its former position. Eberhart and Shi in [58] bounded $x_{i,m}^{(t+1)}$ to avoid a situation whereby particles are spending too much time in infeasible region. A problem dependent fitness function is used in evaluating the superiority of $x_{i,m}^{(t+1)}$. Assuming the present solution is superior to the fitness of $pbest_{i,m}$ or $gbest_m$ then the new position will replace $pbest_{i,m}$ or $gbest_m$ accordingly. Unless the condition for ending the search (either the iteration has reached its peak or we have gotten the desired solution) this updating process will continue. The optimal solution is the best particle found when the stopping criterion is satisfied [59]. In the Asynchronous PSO (APSO), $pbest_{i,m}$ or $gbest_m$ of a particle, its velocity and position are updated immediately after computing their fitness and, as a consequence, they update it having incomplete or imperfect information about the neighbourhood [60]. This result into varieties in the swarm since some of the information is from the previous iteration while some is from the current iteration. In [61], Luo and Zhang (2006), they used the bench-mark functions of Rosenbrock (unimodal) and Griewank (multimodal) to do a performance comparison of SPSO and APSO on the Rosenbrock (unimodal) and Griewank (multimodal) bench-mark functions. They found out that APSO performs better and has a faster convergence than SPSO. Perez and Basterrechea [62] opined from the results of their experiments that APSO is able to find solutions faster and with a similar accuracy as SPSO. They concluded that APSO provides the best accuracy at the expense of computational time.

IV. PRIMAL DUAL INTERIOR POINT METHOD

The primal-dual interior-point (PDIP) method is an excellent example of an algorithm that uses the constraint-reduction methods. Mehrotra in [63] developed the Mehrotra's Predictor-Corrector PDIP algorithm, which has been executed in the majority of the interior-point software suite for solving both linear and convex-conic problems [64]. The primal-dual methods are a new category of interior-point methods that have of recent been practically employed for solving large-scale nonlinear optimization problems according to [67]. Contrary to the traditional primal method, primal-dual method evaluates both the primal variables x and dual Lagrange multipliers λ relating to the constraints concurrently. The disconcerted Karush-Kuhn-Tucker (KKT) equations below can be solved using the precise primal-dual solution ($\mathbf{x}^*_\mu, \lambda^*_\mu$) at a given parameter μ

$$\begin{cases} \nabla F(x) - C^T \lambda = 0 \\ \lambda_i C_i(x) = \mu, i = 1, \dots, m \end{cases}$$

with the constraint $(C(x), \lambda) \geq 0$.

The Newton's algorithm and line search approach are employed to recursively solve any primal or primal-dual sub-problems for a given μ value as stated in [65, 66]. Feasibility and convergence is enforced in the algorithm by selecting the size of step in the iteration. This can be achieved by appropriately reducing the merit function used in gauging degree of advance to the solution. The dual variables of the primal dual can be protected by using Fμ as a function that can incorporate the primal and dual variables [67]; and at the same time measures the harmony between data and the fitting model for a particular choice of the variables [68]. The major setback of the barrier functions is the ineffectiveness of traditional line exploration methods thereby necessitating the development of more efficient line search [69]. According to them, primal-dual method can efficiently handle large linear programming problems (the bigger the problem size the more noticeable the efficiency of the primal dual algorithm). The algorithm is not susceptible to degradation and the number of iterations does not depend on the number of vertices in the feasible search space [70]. Primal-dual algorithm uses considerably less iteration compared to the simplex method and the algorithm is able to generate the ideal solutions for a linear programming problem in less than 100 iterations irrespective of the huge number of variables involved in nearly all its implementations [70]. However, Primal-dual method is hindered by its inability to detect the possibility of having unbounded status of the problem (to a certain extent, the method is labeled as incomplete). This issue has been addressed sufficiently using undiversified model as suggested in [71]. In addition, the computational cost for each iteration in primal-dual is higher than that of the simplex algorithm. Despite this issue, primal-dual method is able to handle a large linear programming problem better than the simplex algorithm. This is because the total work required in providing solution to a large linear programming problem comprised of the multiplication of the number of iterations and the work executed for each iteration [72].

V. PRIMAL DUAL ASYNCHRONOUS PARTICLE SWARM OPTIMIZATION

In one of our previous work, we proposed a new algorithm called Primal-Dual PSO (*pdPSO*) in [73]. In the bid to improve the performance of our algorithm, we developed the Primal-Dual APSO (*pdAPSO*) and used the algorithm to solve flocking problem in swarm robotics [74]. Having discovered from our findings in [75] that APSO have a superior performance compared to the conventional PSO, our intention was to present a hybridized APSO and Primal Dual algorithm which will perform better other earlier variants of PSO that have been developed. The flowchart of the *pdAPSO* algorithm is shown in figure 1 below.

A. Benchmark functions used for performance comparison of primal-dual-APSO (pdAPSO) and state of the art algorithms

In this section we present a comparison between the two new algorithms that we proposed in this paper. We went further to conduct more thorough experiments to evaluate the efficiency of the *pdPSO* algorithm. The test functions that we used for the experiments are as shown in the table 1 below. Thirteen benchmark functions are used in our experiment to further affirm the genuineness of *pdPSO* algorithm. A concise description of these benchmark functions are enumerated in table 1 below. Our reason for adopting these benchmark functions is because they have been generally accepted as suitable functions in measuring the performance of global optimisation algorithms [76, 77, 78, 79]. We made use of twelve functions from the list of functions used in [80]. Based on the attributes of these functions, they can be categorised into three groups. Category one comprises of three (3) unimodal functions. Category two is made up of four composite multimodal functions. The category three comprises of six functions out of which four are rotated multimodal while the remaining two are shifted functions.

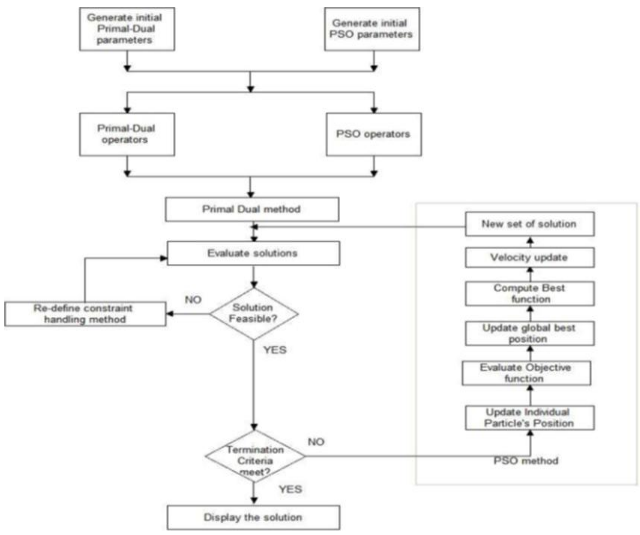

Figure 1: Primal- Dual-APSO (*pdAPSO*) algorithm.

Table 1: Test functions used in the comparisons

Function Name	**Dimension (D)**	**Global opt**	**Search Range**	**Initialization Range**
Unimodal				
Sphere	30	$\{0\}^D$	$[-100,100]^D$	$[-100,50]^D$
Schwefel's P2.22	30	$\{0\}^D$	$[-10,10]^D$	$[-10,10]^D$
Rosenbrock	30	$\{0\}^D$	$[-10,10]^D$	$[-10,10]^D$
Multimodal				
Rastrigin	30	$\{0\}^D$	$[-5.12,5.12]^D$	$[-5.12,5.12]^D$
Ackley	30	$\{0\}^D$	$[-32.768,32.768]^D$	$[-32.768,32.768]^D$
Schwefel	30	$\{0\}^D$	$[-500,500]^D$	[-500,500]D
Griewank	30	$\{0\}^D$	$[-600,600]^D$	$[-600,600]^D$
Rotated and Shifted				
Rotated Rosenbrock	30	$\{0\}^D$	$[-10, 10]^D$	$[-10, 10]^D$
Rotated Rastrigin	30	$\{0\}^D$	$[-5.12,5.12]^D$	$[-5.12,5.12]^D$
Rotated Ackley	30	$\{0\}^D$	$[-32.768,32.768]^D$	$[-32.768,32.768]^D$
Rotated Griewank	30	$\{0\}^D$	$[-600, 600]^D$	$[-600,600]^D$
Shifted Rosenbrock	30	$\{0\}^D$	$[-100, 100]^D$	$[-100,100]^D$
Shifted Rastrigin	30	$\{0\}^D$	$[-100,100]^D$	$[-100,100]^D$

Some performance metrics were used to evaluate the performance of the *pdPSO* in order to know the dependability of the algorithm and the quality of the solution generated. Such performance metrics include the value of mean fitness and standard deviation. The speed of convergence is measured by computing the average number of FEs needed to arrive at a satisfactory solution among successful runs. The dependability of the algorithm is evaluated based on the mean success rate (SR %). The computation of the Mean value of FEs is done only for the successful runs. The ratio of trial runs expressed as a fraction of 100 that successfully reach the standard accuracy is called the success rate. According to Auger and Hansen [81], some algorithms may fail to attain the satisfactory solution for each run on some problems. Another standard of measurement is called success performance (SP).

Where

Success =	(Fit(x*) + (1.0E-5))
X* =	Theoretical global optimal solution
NFE =	Average number of function evaluation required to find solution when all 30 runs are successful.
SP =	(Mean FEs)/(SR%)

B. Performance comparison of pdAPSO algorithms with the state-of-the art PSO variants

In this section we compared the performance of pdAPSO with nine (9) state of the art algorithms as listed in the tables 2-7 below. The conventional PSO algorithm that has been popularly applied in different field is PSO-LDIW which was proposed by [82]. The comprehensive learning strategy PSO (CLPSO) was proposed by [83] with the purpose of producing superior performance compared to the existing PSO variants for multimodal functions. The Perturbed particle swarm optimisation for numerical optimisation was (pPSA) was proposed by [84]. The algorithms device a strategy for handling premature convergence by employing a particle updating approach that centres on the idea of perturbed global best particle. The rank based particle swarm optimisation algorithm with dynamic adaptation (PSO_{rank}) was proposed by [85]. The algorithm exploits the collaborative behavior of particles to make a meaningful increase in the efficiency of the conventional PSO algorithm. Zhan et al. in [76] proposed the orthogonal learning PSO (OLPSO-G). This algorithm uses a perpendicular learning approach to create a favourable and effective model to pilot particles to move in most suitable directions. Huang et al. in [77] developed the Example-based learning PSO (ELPSO) for continuous optimisation. Their purpose is to use example-based learning scheme to proffer a superior performance for multimodal functions. An adaptive parameter tuning of PSO centered on velocity information (APSO-VI) algorithm was proposed by [86]. Diversity enhanced PSO with neighbourhood (DNSPSO) was presented by [87]. This algorithm engages the variety improving method and neighborhood search tactics to attain a swapping between exploration and exploitation. Multiobjective sorting-based learning PSO for continuous optimisation (MLPSO) proposed by [80] uses the MSL approach to direct particles to move in the most suitable path by creating a direction paradigm that have superior fitness value and variety in swarm population. The parameter settings for these PSO variations are specified in Table 9 with reference to their references. The purpose of using these PSO variants for our comparisons is because they are state of the art PSO algorithms which cover a broad period of time from 1999 to 2016. Furthermore, they have been described in literature as high performing variants of PSO with reference to their experimented problems.

Table 2: PSO variants used for our comparative studies.

PSO variants	Parameter Setting	Reference
PSO-LDIW	w : 0.9–0.4, c1 = c2 =2	Shi and Eberhart (1999)
CLPSO	w : 0.9–0.4, c = 1.49, m =7	Liang et al. (2006)
pPSA	w = 0.9, c_1 = 0.5, c_2 = 0.3, σ_{max} = 0.15, σ_{min} = 0.001, $\propto$ = 0.5	Zhao (2010)
PSO_{rank}	w is non-linear, $\propto$ = 0.45, β = 0.385, m = 2	Akbari and Ziarati (2011)
OLPSO-G	w: 0.9–0.4, c = 2.0, G = 5, V_{max} = 0.2 9 x Range	Zhan et al. (2011)
ELPSO	w = 0.729, c = 1.49, m = 4	Huang et al. (2012)
APSO-VI	w:0.9–0.3, c_1 = c_2 = 1.49	Xu (2013)
DNSPSO	w = 0.729, c1 = c2 = 1.49618, k = 2, p_r = 0.9, p_{ns} = 0.6	Wang et al. (2013)
MLSPSO	w:0.9–0.4, c_1 = c_2 = 2	Gang et al (2016)
CDIWPSO	W_1=0.9, W_2=0.4, c_1 = c_2 = 2.0	Feng et al (2007)
CRIWPSO	W_1=0.9, W_2=0.4, c_1 = c_2 = 2.0	Feng et al (2007)
pdAPSO	w:0.9–0.4, c_1 = c_2 = 1.494	

From the experiments that were conducted, the algorithm configurations of *pdAPSO* are as follows. For the PSO part, the inertia weight w is linearly decreasing from 0.9 to 0.4, and c1 and c2 are set to 1.494. The initial value of the maximum centering parameter sigmamax is 0.5. The maximum forcing number etamax is set to 0.25. The Primal Dual segment of the algorithm uses the minimum barrier parameter (mumin) whose initial value is set to 1e-9. The maximum step size (alphamax) is 0.95. The minimum step size (alphamin) is 1e-6. The granularity of backtracking during the search is set to 0.75. And the amount of actual decrease we will accept during backtracking is given the value of 0.01. For a fair comparison among all the PSO variants, the population size is set at 50 and the maximum fitness evaluations (FEs) is set at 30,000. We carried out experiment 30 times for each algorithm using twelve (12) benchmarking functions and the statistical values of the Best Fitness, Worst Fitness, Mean Fitness, Standard Deviation, SP, Success Rate (%), R Runtime (s), and NFE are used in the evaluations.

C. Performance Evaluation Comparison (PEC) on superiority of results

We make comparison of the performance of the PSO algorithms listed in table 2 with that of *pdAPSO*. The results of our comparison are in tables 3 and 4 where we compared the mean and standard deviations of the twelve (12) algorithms. Our performance evaluation of these algorithms is based on the Mean and Standard Deviation. The Mean and Standard Deviations are the Mean value and the Standard Deviation of the best fitness solutions generated by conducting the experiment 1000 times autonomously. The Mean signifies the accuracy of the solutions generated by the algorithm within the given iterated times, and it also indicates the algorithm's convergence velocity. The Standard Deviation reveals the stability and robustness of the algorithm. The best results obtained among the other eleven algorithms that we evaluated their performances are boldfaced. The first three functions (Sphere, Schwefel's P2.22, and Rosenbrock) we considered are unimodal functions. The first two are comparatively easy and virtually all the algorithms can solve them. The two algorithms that proffer the best results for Sphere are *pdAPSO* and ELPSO while *pdAPSO* and OLPSO-G proffers the best results for Schwefel's P2.22. For Rosenbrock function, *pdAPSO* and MSLPSO proffers the best solution. This function is used to test the ability of an algorithm to solve a hard problem because it contains very narrow valley in its landscape. It is only these two algorithms that were able to escape being trapped in its local optima.

Table 3: Mean and Standard Deviation comparisons among twelve (12) PSO algorithms

Algorithm	PEC	f1	f2	f3	f4	f5	f6	f7
PSO-LDIW	Mean	4.68E-23	4.08E-09	5.59E+01	1.85E+01	1.04E-04	2.98E+03	1.84E-04
	Std Dev	8.33E-23	1.09E-09	3.83E+01	2.72E+01	3.85E-03	9.29E+02	2.77E-04
CLPSO	Mean	5.23E-14	2.81E-07	2.25E+01	3.98E-08	3.00E-11	3.86E-03	2.88E-09
	Std Dev	3.66E-14	3.57E-07	1.21E+01	4.54E-08	2.97E-12	4.19E-03	6.47E-08
pPSA	Mean	2.76E-07	2.35E-09	3.57E+01	3.07E-03	8.92E-07	2.58E+03	9.02E-06
	Std Dev	5.93E-07	4.94E-09	2.56E+01	6.25E-02	7.92E-07	6.23E+02	7.29E-06
PSOrank	Mean	3.91E-09	3.73E-12	4.44E+01	1.08E-12	7.82E-10	2.32E+03	1.54E-04
	Std Dev	7.87E-09	5.22E-11	3.14E+01	9.74E-11	5.91E-10	5.12E+02	4.24E-03
OLPSO-G	Mean	6.21E-52	3.77E-28	2.51E+01	8.25E-02	1.33E-14	6.34E+02	3.41E-03
	Std Dev	2.19E-52	6.77E-28	1.77E+01	4.81E-02	5.11E-14	8.09E+01	1.03E-03
ELPSO	Mean	**3.38E-94**	8.08E-24	1.78E+01	2.89E-14	7.69E-15	6.56E-03	9.78E-23
	Std Dev	**1.22E-94**	2.99E-24	1.59E+01	3.18E-14	8.44E-14	3.24E-03	3.11E-23
APSO-VI	Mean	1.37E-12	4.66E-14	1.50E+01	3.82E+00	6.77E-14	2.43E+01	4.88E-12
	Std Dev	8.39E-12	1.44E-14	1.23E+01	4.69E+00	7.35E-14	9.49E+00	2.07E-11
DNSPSO	Mean	8.27E-85	7.97E-26	7.38E+00	7.66E-15	1.89E-14	5.95E+00	3.96E-38
	Std Dev	3.69E-85	5.99E-26	8.82E+00	3.38E-15	2.17E-14	7.17E+00	2.31E-38
MSLPSO	Mean	2.73E-82	1.35E-16	2.90E-01	**2.37E-15**	**7.23E-16**	9.36E+00	**5.91E-43**
	Std Dev	1.69E-82	2.98E-16	3.72E-01	**1.44E-15**	**2.94E-16**	5.44E+00	**1.38E-43**

pdAPSO	Mean	1.13E-29	**1.24E-28**	**2.32E-01**	1.23E+00	1.40E+00	**7.53E-07**	2.83E-03
	Std Dev	2.10E-29	**2.23E-28**	**3.69E-01**	2.51E+00	5.01E+00	**4.06E-06**	3.37E-03

Table 3 continued

Algorithm	PEC	f8	f9	f10	f11	f12	f13
PSO-LDIW	Mean	8.89E+01	9.26E+01	9.55E+00	9.68E-02	6.15E+02	-1.38E+02
	Std Dev	7.27E+01	8.73E+01	7.67E+00	6.33E-01	9.58E+01	3.65E+01
CLPSO	Mean	5.27E+01	4.23E+01	7.21E-05	4.20E-08	4.87E+02	-3.07E+02
	Std Dev	3.88E+01	3.78E+01	6.24E-04	2.72E-09	3.36E+01	7.44E+00
pPSA	Mean	6.06E+01	6.78E+01	6.34E-04	5.14E-04	5.29E+02	-2.78E+02
	Std Dev	5.25E+01	5.33E+01	9.35E-04	3.23E-04	8.53E+01	1.39E+00
PSOrank	Mean	5.87E+01	5.24E+01	3.27E-06	1.52E-03	5.62E+02	-2.42E+02
	Std Dev	6.29E+01	4.19E+01	8.47E-05	2.99E-03	7.87E+01	1.84E+00
OLPSO-G	Mean	3.46E+01	2.81E+01	2.93E-13	4.08E-03	4.63E+02	-3.26E+02
	Std Dev	3.82E+01	2.21E+01	1.79E-12	3.88E-03	4.63E+01	2.33E+00
ELPSO	Mean	3.13E+01	1.62E+00	9.73E-14	5.06E-13	4.26E+02	-3.03E+02
	Std Dev	2.44E+01	3.92E-01	**2.04E-14**	4.24E-13	4.04E+01	5.66E+00
APSO-VI	Mean	4.51E+01	1.23E+01	4.85E-05	1.35E-06	4.41E+02	-2.89E+02
	Std Dev	1.78E+01	8.24E+00	5.25E-05	1.03E-05	3.52E+01	9.24E+00
DNSPSO	Mean	2.92E+01	6.45E-14	5.89E-14	3.98E-21	4.48E+02	-3.11E+02
	Std Dev	2.13E+01	7.19E-14	4.75E-14	4.34E-22	3.27E+01	7.17E+00
MSLPSO	Mean	6.38E+00	**5.89E-15**	**8.68E-14**	**2.17E-35**	4.13E+02	-3.22E+02
	Std Dev	5.45E+00	**8.82E-15**	7.34E-14	**8.92E-34**	3.11E+01	6.34E+00
pdAPSO	Mean	**3.53E+00**	3.30E+00	5.19E-09	2.79E-13	**3.73E+01**	**-3.24E+00**
	Std Dev	**3.11E+00**	2.94E+00	1.47E-08	1.45E-13	**2.69E+00**	**4.32E+00**

For all the experiments we carried out on the multimodal benchmark functions, the Primal Dual method provides PSO the capacity to explore the search space better and exploit the particle in the swarm to its advantage thereby producing enhanced fitness value and create diversity in the swarm population. The mean and standard deviation show that our proposed method can generate better optimum fitness values for some benchmark functions. Also, the convergence accuracy and convergence velocity of our proposed method is commendable. The standard deviation specifies that the divergence of the optimum fitness result that our proposed technique produced is relatively less. This is a prove that our proposed method possesses better stability and robustness. It is anticipated that *pdAPSO* will escape from being trapped in local minima and produce superior results on multimodal functions. For the all the functions tested *pdAPSO* converged to the global optimum. Our proposed algorithm produced the best result for Schwefel, Rosenbrock, Griewank, Rotated Rosenbrock, Shifted Rosenbrock, and Shifted Rastrigin functions. The results of our tests demonstrated that *pdAPSO* possess that ability to effectively handle premature convergence problem and escape from being trapped in local minima on majority of the multimodal functions. The successful attainment of global optima solutions on many of the multimodal functions indicates that the performances of *pdAPSO* algorithm have really been enhanced through the fusion of Primal-Dual method and PSO algorithm.

We also investigated the performances of the eleven algorithms on rotated and shifted functions. Rotated Rosenbrock, Rotated Rastrigin, Rotated Ackley, and Rotated Griewank are multimodal functions with rotated coordinates. The *pdAPSO* algorithm attained global optima for all the rotated functions. On Rotated Rosenbrock, Shifted Rosenbrock, and Shifted Rastrigin functions, *pdAPSO* produced the best result. MSLPSO achieved the best result for Griewank, Rastrigin, Ackley, Rotated Rastrigin, and Rotated Griewank functions. ELPSO generated the best result for the Sphere function. It should be noted that the rotation does not affect the performance of the *pdAPSO*. Infact the effectiveness of our algorithms become more pronounced with test on all the rotated functions especially Rotated Rosenbrock where *pdAPSO* produced the most accurate result and closely followed by MSLPSO. To be precise, our experiments confirmed the observation of Wang et al. [87] that the Rotated Rosenbrock function proved very difficult for other PSO algorithms to escape being trapped in its local optima as the function becomes more problematic after rotating its coordinates.

The outcome of our experiments also indicates that *pdAPSO* competes very well with other state of the art algorithms. On the Shifted Rosenbrock functions, *pdAPSO* produced the best result and closely followed by MSLPSO. The other PSO algorithms are trapped in local optima this function. In summary, the rotation and shift affected the performance of the other nine algorithms while the efficiency of *pdAPSO* becomes more noticeable with the rotation and the shift. The comparisons reveal that the integration of Primal-Dual into PSO is advantageous to enhancing the performance of PSO. We hereby conclude that *pdAPSO* have a superior performance compared to the other PSO variants on one of the rotated functions and on the two shifted functions.

D. Performance Evaluation Comparison (PEC) on the dependability and speed of convergence

The dependability of an algorithm is determined by the mean of success rate on the entire test functions. The success rate (SR) is the rate of the optimum fitness result in the criterion range experimenting

1000 times independently. The SR indicates the algorithm's global search competence. The convergence speed in attaining the global optimum is also a striking standard for determining the performance of any optimisation algorithm. The rates of success of all eleven variants of PSO algorithm on individual test function and the dependability of the algorithms are shown in Table 4 below.

Table 4: Performance Evaluation Comparison (PEC) of the dependability and speed of convergence

Algorithm	PEC	f1	f2	f3	f4	f5	f6	f7
PSO-LDIW	**Mean Fes**	5571	12,049	8938	12,164	11,216	18,243	7781
	SR%	**100**	**100**	**100**	**100**	16.67	30.33	40
	SP	5571	12,049	8938	12,164	67,323	60,148	19,452
CLPSO	**Mean FEs**	7069	14,702	6786	15,423	12,957	10,024	9204
	SR%	**100**	**100**	**100**	**100**	**100**	**100**	**100**
	SP	7069	14,702	6786	15,423	12,957	10,024	9204
pPSA	**Mean FEs**	6954	11,205	10,080	11,835	13,720	15,045	13,567
	SR%	**100**	**100**	**100**	**100**	**100**	43.33	**100**
	SP	6954	11,205	10,080	11,835	13,720	34,722	13,567
PSOrank	**Mean FEs**	6631	9929	9607	10,387	13,142	13,956	8318
	SR%	**100**	**100**	**100**	**100**	**100**	76.67	**100**
	SP	6631	9929	9607	10,387	13,142	18,205	8318
OLPSO-G	**Mean FEs**	3872	9301	11,680	9629	10,751	9827	7432
	SR%	**100**	**100**	**100**	**100**	**100**	**100**	63.33
	SP	3872	9301	11,680	9629	10,751	9827	11,735
ELPSO	**Mean FEs**	4396	8973	9287	9302	6698	9306	7672
	SR%	**100**	**100**	**100**	**100**	**100**	**100**	**100**
	SP	4396	8503	9287	9302	6698	9306	7672
APSO-VI	**Mean FEs**	24,910	18,932	23,940	27,127	29,361	23,194	28,381
	SR%	**100**	**100**	**100**	**100**	**100**	**100**	**100**
	SP	24,910	18,932	23,940	27,127	29,361	23,194	28,381
DNSPSO	**Mean FEs**	4767	10,345	6269	10,073	10,673	10,083	8315
	SR%	**100**	**100**	**100**	**100**	**100**	**100**	**100**
	SP	4767	10,345	6269	10,073	10,673	10,083	8315
MSLPSO	**Mean FEs**	5853	8828	6198	7896	12,992	9362	7159
	SR%	**100**	**100**	**100**	**100**	**100**	**100**	**100**
	SP	5853	8828	6198	7896	12,992	9362	7159
pdAPSO	**Mean FEs**	**332**	**495.00**	**2124**	**3545**	**2124**	**5929**	**1160**
	SR%	**100**	**100**	**100**	**100**	**100**	**100**	**100**
	SP	**332**	**532**	**3321.16**	**3526.79**	**3321.16**	**8316.73**	**1734.56**

Table 4 Continued

Algorithm	PEC	f8	f9	f10	f11	f12	f13
PSO-LDIW	**Mean FEs**	9897	14,278	–	8241	9235	11,097
	SR%	33.33	76.67	**0**	13.33	13.33	26.66
	SP	29,694	18,625	-	61,823	69,280	41,624
CLPSO	**Mean FEs**	10,623	16,085	24,727	12,514	13,024	13,295
	SR%	76.67	**100**	40	**100**	43.33	**100**
	SP	13,857	16,085	61,817	12,514	30,058	13,295
pPSA	**Mean FEs**	14,929	12,537	16,296	9574	9783	11,092
	SR%	63.33	**100**	16.67	60	23.33	53.33
	SP	23,573	12,537	97,815	15,957	47,351	20,799
PSOrank	**Mean FEs**	13,057	11,043	15,292	8933	12,312	10,551
	SR%	76.67	**100**	80	43.33	16.67	63.33
	SP	17,032	11,043	19,115	20,616	73,902	16,660
OLPSO-G	**Mean FEs**	12,958	10,074	12,707	9404	10,331	9012
	SR%	93.33	**100**	**100**	33.33	66.67	**100**
	SP	13,884	10,074	12,707	28,215	15,496	9012
ELPSO	**Mean FEs**	10,034	9737	9318	9242	10,376	10,075
	SR%	**100**	**100**	**100**	**100**	73.33	**100**
	SP	10,034	9737	9318	9242	14,149	10,075
APSO-VI	**Mean FEs**	27,035	11,003	27,824	297,10	26,292	28,039
	SR%	73.33	**100**	36.67	**100**	83.33	76.67
	SP	36,868	11,003	75,897	29,710	31,552	36,575
DNSPSO	**Mean FEs**	9836	9265	13,239	5981	8832	9923
	SR%	**100**	**100**	**100**	**100**	86.67	**100**
	SP	9836	9265	13,239	5981	10,192	9923
MSLPSO	**Mean FEs**	8742	8792	9153	7748	9783	12,081
	SR%	**100**	**100**	**100**	**100**	**100**	**100**
	SP	8742	8792	9153	7748	9783	12,081
pdAPSO	**Mean FEs**	**2197**	**5403**	**2185**	**1105**	**2256**	**3402**

	SR%	**100**	**100**	**99.5**	**98.7**	**100**	**100**
	SP	**3380.16**	**7301.35**	**3378.16**	**1860.25**	**3182.72**	**3526.38**

From the results of our experiment depicted in Table 4 above, the success rate of the proposed algorithm is an indication that it has a fairly good global search capacity as it can escape from the local optimum and search out the global optimum result. The mean dependability of *pdAPSO* is very high. This is an indication that the fusion of Primal Dual method and PSO will increase the dependability to PSO in overcoming premature convergence and converging to global optima. It is worthy of note that *pdAPSO* converged to the global optimum for all the test functions. PSO-LDIW was unable to converge on Rotated Ackley function. The ratio of dependability of *pdAPSO* indicated that our algorithm offers a dependable and robust method for providing solution to global optimisation problems. The pace at which an algorithm attains the global optimum is a very important parameter for assessing the performance of the algorithm. Since the Primal Dual method is a robust optimisation algorithm, it is expected that *pdAPSO* will produce superior result in comparison to so other state of the art algorithm with a better speed of convergence. To substantiate our claim, the results of Mean FEs and SP, for the eleven algorithms are shown in Table 4. The best results are boldface in each of the Tables.

It is very obvious from those tables that the speed of convergence of *pdAPSO* algorithm is superior to the other PSO algorithms on all the benchmark functions. For instance, on Schwefel's function, the mean FEs of 5571, 7069, 6954, 6631, 3872, 4396, 24,910, 4767, and 5853 are required by PSO-LDIW, CLPSO, pPSA, PSOrank, OLPSO-G, ELPSO, APSO-VI, DNSPSO and MSLPSO respectively to attain the global optima. However, *pdAPSO* only uses 332 which is an indication that *pdAPSO* is the fastest. To be concise, *pdAPSO* uses the lowest number of FEs to attain satisfactory solutions for all the 13 benchmark functions. This is another confirmation that the Primal Dual method has enhanced the PSO algorithm in producing better fitness value and creating diversity in the swarm population to improve the convergence speed of PSO particles.

VI. CONCLUSION

This paper presents a new hybrid optimization algorithm named Primal Dual Interior Point Method Asynchronous Particle Swarm Optimization (*pdAPSO*). This algorithm combines the explorative ability of PSO with the exploitative capacity of the Primal Dual Interior Point Method thereby possessing a strong capacity of avoiding premature convergence. Several experiments were conducted. We did a comparison of the performance and superiority of solutions of the 10 algorithms, and the outcome of our tests show that *pdAPSO* have the capacity to overcome the problem premature convergence and prevent particles from being trapped in local minima on many all the functions. The comparison of dependability and speed of convergence of the 10 algorithms on 13 benchmark functions was also done. The result of our experiment shows that *pdAPSO* is reliable and robust algorithm for solving global optimisation problems. The convergence speed of *pdAPSO* algorithm was compared to the other state of the art PSO algorithms and our proposed algorithms proved to attain the global optima on all the benchmark functions in the shortest run time. The behaviour of *pdAPSO* under the unimodal and multimodal functions shows that the algorithm will be a suitable tool in solving complicated optimization problems that PSO alone or Primal Dual alone cannot solve efficiently.

REFERENCES

[1] Venayagomoorthy G. K., Del Valle Y., Mohagheghi .S, Hernandez J. C., and Harley R. G. Particle swarm optimization: basic concepts, variants and applications in power systems. *IEEE Trans Evol Comput, 12 (2)*, 2008, 171 – 195.

[2] Basturk B., and Karaboga D. A powerful and Efficient Algorithm for Numerical Function Optimization: Artificial Bee Colony (ABC) algorithm, *J. Glob Optim; 39*,2007, 459-471.

[3] Kennedy J., Eberhart R. C. Particle swarm optimization, in: *Proceedings of the International Conference on Neural Networks, vol. 4, IEEE Press*, Piscataway, NJ, 1995; 1942-1948.

[4] Guerra F. A., Coelho L. D. S. Applying Particle Swarm Optimization to Adaptive Controller. A. Saad et al. (Eds.): *Soft Computing in Industrial Applications, ASC; 39*, 2007, 82–91.

[5] Eberhart R. C., Shi Y. Parameter Selection in particle swarm optimization, *In Proceedings of the 7th International Conference on Evolutionary Programming*, 1998; 591 – 600.

[6] Rusman H. An Efficient Particle Swarm Optimization Technique for Solving the Non-Convex Economic Dispatch Problems. *International Journal of Engineering Sciences*, 2(5), 2013, 137– 144.

[7] Abraham A., Konar A., and Das S. Particle Swarm Optimization and Differential Evolution Algorithms: Technical Analysis, Applications and Hybridization Perspectives. *Studies in Computational Intelligence, (SCI), 116*, 2008, 1–38.

[8] Abraham A., Pant M., Bouvry P., and Thangaraj R. Particle swarm optimization: Hybridization perspectives and experimental illustrations, *Appl. Math. Comput, 217,* 2011, 5208 – 5226, doi:10.1016/j.amc.2010.12.053.

[9] Torn A. Z., Global Optimization, *Lecture Notes in Computer Science, Springer-Verlag*, 1989; 350.

[10] Laird C. D. *Structured Large-Scale Nonlinear Optimization Using Interior-Point Method: Applications in Water Distribution Systems*. PhD Thesis Department of Chemical Engineering, Carnegie Mellon University, Pittsburgh, Pennsylvania, 2006.

[11] Blohm Winternitz L. M. *Primal-Dual Interior-Point Algorithms for Linear Programs with Many Inequality Constraints*. Dissertation submitted to the Faculty of the Graduate School of the University of Maryland, College Park in partial fulfillment of the requirements for the degree of Doctor of Philosophy, 2010.

[12] Robinson J., Sinton S., Samii Y. R. Particle swarm, genetic algorithm, and their hybrids: optimization of a profiled corrugated horn antenna, in*: Proceedings of the IEEE International Symposium in Antennas and Propagation Society*, 2002; 314–317.

[13] Krink, T., Løvbjerg, M. The lifecycle model: combining particle swarm optimization, genetic algorithms and hill climbers, *Proceedings of the Parallel Problem Solving from Nature*, 2002; 621–630.
[14] Conradie E., Miikkulainen R., Aldrich C. Intelligent process control utilizing symbiotic memetic neuro-evolution, *Proceedings of the IEEE Congress on Evolutionary Computation*, 2002; 623 – 628.
[15] Grimaldi E. A., Grimacia F., Mussetta M., Pirinoli P., and Zich R. E. A new hybrid genetical – swarm algorithm for electromagnetic optimization, in: *Proceedings of International Conference on Computational Electromagnetics and its Applications*, Beijing, China, 2004; 157–160.
[16] Gandelli F. Grimaccia, Mussetta M., Pirinoli P. Zich R. E. Development and validation of different hybridization strategies between GA and PSO, *Proceedings of the IEEE Congress on Evolutionary Computation*, 2007; 2782–2787.
[17] Gandelli Grimaccia F., Mussetta M., Pirinoli P., Zich R. E. Genetical swarm optimization: a new hybrid evolutionary algorithm for electromagnetic application, in: *Proceedings of the 18th International Conference on Applied Electromagnetics, ICECcom*, Dubrovnik, Croatia, 2005; 269–272.
[18] Grimaccia Mussetta M., Pirinoli P., and Zich R. E. Genetical swarm optimization: self-adaptive hybrid evolutionary algorithm for electromagnetic, *IEEE Transactions on Antennas and Propagation, 55 (3)*, 2007, 781–785.
[19] Gandelli Grimaccia F., Mussetta M., Pirinoli P.,and Zich R. E., Genetical swarm optimization: an evolutionary algorithm for antenna design, *Journal of Automatika, 47 (3-4)*, 2006, 105–112.
[20] Hendtlass T. A Combined Swarm differential evolution algorithm for optimization problems, in: *Proceedings of 14th International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems, Lecture Notes in Computer Science, 2070*, Springer Verlag, 2001, 11–18.
[21] Talbi H., and Batouche M. Hybrid particle swarm with differential evolution for multimodal image registration, *Proceedings of the IEEE International Conference on Industrial Technology, 3*, 2004, 1567–1573.
[22] Hao Z. F., Gua G. H., and Huang H. A particle swarm optimization algorithm with differential evolution, in: *Proceedings of Sixth International Conference on Machine Learning and Cybernetics*, 2007; 1031–1035.
[23] Omran M., Engelbrecht A.P., Salman A. Bare bones differential evolution, *European Journal of Operational Research*, 2008, 128–139.
[24] Jose G. N., Alba E., Apolloni J. Particle swarm hybridized with differential evolution: black box optimization benchmarking for noisy functions, in*: Proceedings of International Conference Genetic and Evolutionary Computation*, 2009; 2343–2350.
[25] Zhang C., Ning J., Lu S. Ouyang D., and Ding T. A novel hybrid differential evolution and particle swarm optimization algorithm for unconstrained optimization, *Operations Research Letters, 37*, 2009, 117–122.
[26] Liu H., Cai Z., and Wang Y. Hybridizing particle swarm optimization with differential evolution for constrained numerical and engineering optimization, *Applied Soft Computing, 10*, 2009, 629–640.
[27] Caponio Neri F., and Tirronen V. Superfit control adaption in memetic differential evolution frameworks, *Soft Computing, 13*: 2009, 811–831.
[28] Xu W., and Gu X. A hybrid particle swarm optimization approach with prior crossover differential evolution, in: *Proceedings of ACM/SIGEVO Summit on Genetic and Evolutionary Computation*, 2009; 671–678.
[29] Pant, M. Thangaraj, R., and Abraham, A. DE-PSO: a new hybrid meta-heuristic for solving global optimization problems, *New Mathematics and Natural Computation*, 2009.
[30] Khamsawang S., and Wannakarn P. Jiriwibhakorn S. Hybrid PSO-DE for solving the economic dispatch problem with generator constraints, in*: Proceedings of the IEEE International Conference on Computer and Automation Engineering*, 2010, 135–139.
[31] Shelokar P.S., Siarry P., Jayaraman V. K., and Kulkarni B. D. Particle swarm and ant colony algorithms hybridized for improved continuous optimization, *Applied Mathematics and Computation*, 188, 2007, 129–142.
[32] Hendtlass, T., and Randall, M. A survey of ant colony and particle swarm meta-heuristics and their application to discrete optimization problems, in: *Proceedings of the Inaugural Workshop on Artificial Life*, 2001; 15–25.
[33] Victoire, T. A. A., and Jeyakumar, A. E. Hybrid PSO–SQP for economic dispatch with valve-point effect, *Electric Power Systems Research*, 71, 2004, 51–59.
[34] Boggs P.T., and Tolle J. W. Sequential quadratic programming, *Acta Numerica, vol. 4,* Cambridge University Press, Cambridge, 1995; 1–52.
[35] Grosan Abraham A., and Nicoara, M. Search optimization using hybrid particle sub-swarms and evolutionary algorithms, *International Journal of Simulation Systems, Science and Technology, 6*, 2005, 60–79.
[36] Liu, H., Abraham, and A., Zhang, W. A fuzzy adaptive turbulent particle swarm optimization, *International Journal of Innovative Computing and Applications,* 1, 2007, 39–47.
[37] Sha D.Y., and Hsu C. Y. A hybrid particle swarm optimization for job shop scheduling problem, *Computers & Industrial Engineering, 51*, 2006, 791–808.
[38] He Q., and Wang L. 2007. A hybrid particle swarm optimization with a feasibility-based rule for constrained optimization, *Applied Mathematics and Computation, 186,* 2007, 1407–1422.
[39] Mo W., and Guan S. U., Puthusserypady S. K. A novel hybrid algorithm for function optimization: particle swarm assisted incremental evolution strategy, *Studies in Computational Intelligence, 75*, 2007, 101–125.
[40] Fan S. K., and Zahara E. A hybrid simplex search and particle swarm optimization for unconstrained optimization, *European Journal of Operational Research, 181*, 2007, 527–548.
[41] Fan S. K., Liang Y. C., and Zahara E. Hybrid simplex search and particle swarm optimization for the global optimization of multimodal functions, *Engineering Optimization, 36*: , 2004, 401–418.
[42] Zahara Y., and Kao T. Hybrid Nelder–Mead simplex search and particle swarm optimization for constrained engineering design problems, *Expert Systems with Applications, 36*, 2009, 3880 – 3886.
[43] Guo Q. J., Yu H. B., and Xu A. D. A hybrid PSO-GD based intelligent method for machine diagnosis, *Digital Signal Processing, 16*, 2006, 402–418.
[44] Shen Q., Shi W. M., and Kong, W. Hybrid particle swarm optimization and tabu search approach for selecting genes for tumor classification using gene expression data, *Computational Biology and Chemistry,32,* 2007, 53–60.
[45] Ge H. W., Sun L., Liang Y. C., and Qian F. An effective PSO and AIS-based hybrid intelligent algorithm for job-shop scheduling, *IEEE Transactions on Systems, Man, And Cybernetics - Part A: Systems and Humans, 38*, 2008, 358–368.
[46] Song S., Kong L., Gan Y., and Su R. Hybrid particle swarm cooperative optimization algorithm and its application to MBC in alumina production, *Progress in Natural Science, 18*, 2008, 1423–1428.
[47] Kao Y. T., Zahara E., Kao I. W. A hybridized approach to data clustering, *Expert Systems with Applications, 34*, 2008, 1754–1762.
[48] Murthy R., Arumugam M. S., and Loo C. K. Hybrid particle swarm optimization algorithm with fine tuning operators, *International Journal of Bio-Inspired Computation, 1*, 2009, 14–31.

[49] Kuo I. H., Horng S. J., Kao T. W., Lin T. L., Lee C. L., Terano T., and Pan Y. An efficient flow shop scheduling algorithm based on a hybrid particle swarm optimization model, *Expert Systems with Applications*, *36*, 2009, 7027–7032.
[50] Chen M. R., Li X., Zhang X., and Lu Y Z. A novel particle swarm optimizer hybridized with extremal optimization, *Applied Soft Computing*, *10*, 2010, 367– 373.
[51] Boettcher S., and Percus A. G. Extremal optimization: methods derived from coevolution, in: *Proceedings of the Genetic and Evolutionary Computation Conference*, 1999; 825–832.
[52] Kavehand S. Talatahari. Particle swarm optimizer, ant colony strategy and harmony search scheme hybridized for optimization of truss structures, *Computer Structure, 87*, 2009a, 267 – 283.
[53] Kavehand S. Talatahari. A particle swarm ant colony optimization for truss structures with discrete variables, *Constructional Steel Research*, *65*, 2009b, 1558–1568.
[54] Wei Z. He, and Pei W. Swarm directions embedded in fast evolutionary programming, in: *Proceedings of the World Congress on Computational Intelligence*, *2*, 2002, 1278–1283.
[55] Pant M., Thangaraj R., and Abraham A. Particle swarm optimization using adaptive mutation, in: *Proceedings of 19th International Conference on Database and Expert Systems Application*, Italy, 2008; 519–523.
[56] Tang Q., and Eberhard P. A PSO-based algorithm designed for a swarm of mobile robots, *Journal of Industrial Application*. *Struct Multidisc Optim.*, *44*, 2011, 483-498.
[57] Sedlaczek K., and Eberhard P. Constrained Particle Swarm Optimization of Mechanical Systems, *6th World Congresses of Structural and Multidisciplinary Optimization*, Rio de Janeiro, 30 May - 03 June, Brazil, 2005.
[58] Kennedy J., Eberhart R. C., and Shi Y. *Swarm Intelligence*, Morgan Kaufmann Publishers Inc. San Francisco, CA, USA, 2001.
[59] Aziz N. A. A., and Ibrahim Z. Asynchronous Particle Swarm Optimization for Swarm Robotics, International Symposium on Robotics and Intelligent Sensors (IRIS 2012). *Procedia Engineering*, *41*, 2012, 951 – 957.
[60] Eberhart, R. C., and Shi, Y. Comparing Inertia Weights and Constriction Factors in Particle Swarm Optimization, In*: Proceedings of the Congress on Evolutionary Computation, 1*, 2001, 84 – 88.
[61] Luo, J., and Zhang, Z. Research on the Parallel Simulation of Asynchronous Pattern of Particle Swarm Optimization, *Computer Simulation, 22(6)*, 2006, 78–70.
[62] Perez, J. R., and Basterrechea, J. Particle swarm optimization and its application to antenna far field-pattern prediction from planar scanning, *Microwave and optical technology letters, 44(5)*, 2005, 398–403.
[63] Mehrotra S. On the implementation of a primal-dual interior point method, *SIAM Journal on Optimization, 2*, 1992, 575-601.
[64] Frisch K. R. *The logarithmic potential method of convex programming*, Technical Report, University Institute of Economics, Oslo, Norway, 1995.
[65] Wright S. J. *Primal-dual interior-point methods*, SIAM Philadelphia, PA, 1st edition, 1997.
[66] Boyd S., Vandenberghe L. *Convex Optimization*, Cambridge University Press, New York, 1st edition, 2004.
[67] Quintana V. H., Torres G. L. Introduction to Interior-Point Methods, *IEEE PES task Force on Interior-Point Methods Applications to Power Systems,* Available online at http://wathvdc3.uwaterloo.ca/~iee-ipm, 1997.
[68] Sofer A., Johnson C. A., Seidel J. Interior-point methodology for 3-D PET reconstruction, *IEEE Trans. Medical Imaging, 19 (4)*, 2000, 271-285.
[69] Chouzenoux E., Idier J., and Moussaoui S. Efficiency of Line Search Strategies in Interior Point Methods for Linearly Constrained Signal Restoration, in*: Proceedings of the IEEE Workshop on Statistical Signal Processing (SSP),* Nice, France, 28-30 June 2011, 101-104.
[70] Glavic M., and Wehenkel L. *Interior Point Methods: A Survey, Short Survey of Applications to Power Systems, and Research Opportunities*, Technical Report. University of Liège Electrical Engineering and Computer Science Department Sart Tilman B-28 4000 Liege, Belgium, 2004.
[71] Gilbert J. C., Armand P., and Jan-Jégou S. A feasible BFGS interior point algorithm for solving strongly convex minimization problems, *SIAM J. Optimization, 11*, 2000, 199 – 222.
[72] Paul Armand, Jean C. Gilbert, and Sophie Jan-Jégou. A feasible BFGS interior point algorithm for solving strongly convex minimization problems, *SIAM J. Optimization; 11*, 2000, 199 – 222.
[73] Dada, E. G., and Ramlan, E. I. Primal-Dual Interior Point Method Particle Swarm Optimization (pdipmPSO) Algorithm. In: *3rd Int'l Conference on Advances in Engineering Sciences and Applied Mathematics (ICAESAM'2015)*, London (UK), 2015, 117-124.
[74] Dada, E. G., and Ramlan, E. I. A Hybrid Primal-Dual-PSO (pdipmPSO) Algorithm for Swarm Robotics Flocking Strategy. *The Second International Conference on Computing Technology and Information Management (ICCTIM2015).* Malaysia, IEEE, 2015b, 93-98.
[75] Dada, E. G., and Ramlan, E. I. Understanding the limitations of particle swarm algorithm for dynamic optimization tasks. *ACM Computing Survey, 49(1),* 2016, 1-25. DOI: http://dx.doi.org/10.1145/2906150.
[76] Zhan Z. H, Zhang J, Li Y, and Shi Y. H. Orthogonal learning particle swarm optimisation. *IEEE Trans Evol Comput, 15*, 2011, 832–847.
[77] Huang H, Qin H, Hao Z, and Lim A. Example-based learning particle swarm optimisation for continuous optimisation. *Inf Sci, 182(1)*, 2012, 125–138.
[78] Suganthan P. N, Hansen N, Liang J. J, Deb K, Chen Y. P, Auger A, and Tiwari S. Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimisation. In: *Proceedings of the congress on evolutionary computation*, 2005; 1 – 50.
[79] Liang J. J, Suganthan P. N, and Deb K. Novel composition test functions for numerical global optimisation. In: *Proceedings of the IEEE swarm intelligence symposium*, 2005: 1–14.
[80] Gang Xu, Binbin Liu, Jun Song, Shuijing Xiao, and Aijun Wu. Multiobjective sorting-based learning particle swarm optimisation for continuous optimisation. *Nat Comput*, 2016, DOI :10.1007/s11047-016-9548-3
[81] Auger A, and Hansen N. Performance evaluation of an advanced local search evolutionary algorithm. In: *Proceedings of the IEEE congress on evolutionary computation, 1772*, 2005, 1777–1784.
[82] Shi, Y., and Eberhart, R. C. Parameter Selection in particle swarm optimisation, In *Proceedings of the 7th International Conference on Evolutionary Programming*, 1999: 591 – 600.
[83] Liang, J. J., Qin, A. K., Suganthan, P. N., and Baskar, S. Comprehensive Learning Particle Swarm Optimizer for Global Optimisation of Multimodal Functions, *IEEE transactions on evolutionary computation, 10 (3),* 2006, 281 – 295.
[84] Zhao X. A perturbed particle swarm algorithm for numerical optimization. *Appl Soft Comput., 10 (1)*, 2010, 119–124.
[85] Akbari R., and Ziarati K. A rank based particle swarm optimization algorithm with dynamic adaptation. *J Comput Appl Math, 235*, 2011, 2694–2714.
[86] Xu G. An adaptive parameter tuning of particle swarm optimization algorithm. *Appl Math Comput, 219*, 2013, 4560–4569.

[87] Wang H, Sun H, Li C. H., Rahnamayan S, and Pan J. S. Diversity enhanced particle swarm optimization with neighborhood search. *Inf Sci*, *223*, 2013, 119–135.