An Analysis to Hybrids of Particle Swarm Optimisation and Differential Evolution Algorithm

Ahmad Rezaee Jordehi, Jasronita Jasni

Department of Electrical Engineering, University Putra Malaysia
ahmadrezaeejordehi@gmail.com

Abstract: Particle swarm optimisation (PSO) is one of the most promising optimisation techniques that exposes desirable computational behavior. However, hybridizing it with other optimisation techniques may lead to more efficient algorithms, because by hybridization the constituent techniques reinforce each other's strengths and cover each others' shortcomings. One of the algorithms that its hybridization with PSO leads to encouraging outcomes is differential evolution (DE). This paper presents a comprehensive analysis on various variants which are hybrids of PSO and DE.
[Jordehi R, Jasni J. **An Analysis to Hybrids of Particle Swarm Optimisation and Differential Evolution Algorithm.** *Biomedicine and Nursing* 2018;4(3):82-86]. ISSN 2379-8211 (print); ISSN 2379-8203 (online). http://www.nbmedicine.org. 11. doi:10.7537/marsbnj040318.11.

Keywords: Particle swarm optimisation, Differential Evolution, Hybrid.

1. Introduction

There exist so many optimisation problems in various areas of science and engineering. For solving them, there exist twofold approaches; classical approaches and heuristic approaches. Classical approaches are not efficient enough in solving optimisation problems. Since they suffer from curse of dimensionality and also require preconditions such as continuity and differentiability of objective function that usually are not satisfied.

Heuristic approaches which are usually bio-inspired include a lot of approaches such as genetic algorithms, evolution strategies, differential evolution and so on. Heuristics do not expose most of the drawbacks of classical and technical approaches. Among heuristics, particle swarm optimisation (PSO) has shown more promising behavior.

PSO is a stochastic, population-based optimisation technique introduced by Kennedy and Eberhart (Kennedy and Eberhart 1995). It belongs to the family of swarm intelligence computational techniques and is inspired of social interaction in human beings and animals (especially bird flocking and fish schooling).

Some PSO features that make it so efficient in solving optimisation problems are the followings:

- In comparison with other heuristics, it has less parameters to be tuned by user.
- Its underlying concepts are so simple. Also its coding is so easy.
- It provides fast convergence.
- It requires less computational burden in comparison with most other heuristics.
- It provides high accuracy.
- Roughly, initial solutions do not affect its computational behavior.

Although PSO exposes very desirable computational behavior, hybridizing it with other optimisation techniques may lead to even more efficient algorithms, because by hybridization the constituent techniques reinforce each other's strengths and cover each others' shortcomings. This paper presents an analysis on various variants which are hybrids of PSO and DE operators. It discusses thoroughly about each variant, its characteristics, advantages and disadvantages. The paper is organised as follows; in section II, an overview of PSO is presented. In section III, an exhaustive analysis of hybrid PSO-DE variants is provided. Finally, conclusions are drawn in section IV.

2. Basic Concepts and Variants of PSO

PSO starts with the random initialisation of a population (swarm) of individuals (particles) in the n-dimensional search space (n is the dimension of problem in hand). The particles fly over search space with adjusted velocities. In PSO, each particle keeps two values in its memory; its own best experience, that is, the one with the best fitness value (best fitness value corresponds to least objective value since fitness function is conversely proportional to objective function) whose position and objective value are called P_i and P_{best} respectively and the best experience of the whole swarm, whose position and objective value are called P_g and g_{best} respectively. Let denote the position and velocity of particle i with the following vectors:

$$X_i = (X_{i1}, X_{i2}, \dots, X_{id}, \dots, X_{in})$$

$$V_i = (V_{i1}, V_{i2}, \dots, V_{id}, \dots, V_{in})$$

The velocities and positions of particles are updated in each time step according to the following equations:

$$V_{id}(t+1) = V_{id}(t) + C_1 r_{1d}(P_{id} - X_{id}) + C_2 r_{2d}(P_{gd} - X_{id}) \quad (1)$$

$$X_{id}(t+1) = X_{id}(t) + V_{id}(t+1) \quad (2)$$

Where C_1 and C_2 are two positive numbers and r_{1d} and r_{2d} are two random numbers with uniform distribution in the interval [0,1]. Here, according to (1), there are three following terms in velocity update equation:

1) The first term this models the tendency of a particle to remain in the same direction it has traversing and is called "inertia," "habit," or "momentum."

2) The second term is a linear attraction toward the particle's own best experience scaled by a random weight $C_1 r_{1d}$. This term is called "memory," "nostalgia," or "self-knowledge."

3) The third term is a linear attraction toward the best experience of the all particles in the swarm, scaled by a random weight $C_2 r_{2d}$. This term is called "cooperation," "shared information," or "social knowledge."

The procedure for implementation of PSO is as follows:

1) Particles' velocities and positions are Initialised randomly, the objective value of all particles are calculated, the position and objective of each particle are set as its P_i and P_{best} respectively and also the position and objective of the particle with the best fitness (least objective) is set as P_g and g_{best} respectively.

2) Particles' velocities and positions are updated according to equations (1) and (2).

3) Each particle's P_{best} and P_i are updated, that is, if the current fitness of the particle is better than its P_{best}, P_{best} and P_i are replaced with current objective value and position vector respectively.

4) P_g and g_{best} are updated, that is, if the current best fitness of the whole swarm is fitter than g_{best} , g_{best} and P_g are replaced with current best objective and its corresponding position vector respectively.

5) Steps 2-4 are repeated until stopping criterion (usually a prespecified number of iterations or a quality threshold for objective value) is reached.

It should be mentioned that since the velocity update equations are stochastic, the velocities may become too high, so that the particles become uncontrolled and exceed search space. Therefore, velocities are bounded to a maximum value V_{max}, that is (R. Eberhart 2001)

$$\text{If } |V_{id}| > V_{max} \text{ then } V_{id} = \text{sign}(V_{id})V_{max} \quad (3)$$

Where sign represents sign function.

However, primary PSO characterised by (1) and (2) does not work desirably; especially since it possess no strategy for adjusting the trade-off between explorative and exploitative capabilities of PSO. Therefore, the inertia weight PSO is introduced to remove this drawback. In inertia-weight PSO, which is the most commonly-used PSO variant, the velocities of particles in previous time step is multiplied by a parameter called inertia weight. The corresponding velocity update equations are as follows (Shi and Eberhart 1998; Shi and Eberhart 1999):

$$V_{id}(t+1) = \omega V_{id}(t) + C_1 r_{1d}(P_i - X_{id}) + C_2 r_{2d}(P_{gd} - X_{id})$$

$$X_{id}(t+1) = X_{id}(t) + V_{id}(t+1) \quad (4)$$

Inertia weight adjusts the trade-off between exploration and exploitation capabilities of PSO. The less the inertia weight is, the more the exploration capability of PSO will be and vice versa. Commonly, it is decreased linearly during the course of the run, so that the search effort is mainly focused on exploration at initial stages and is focused more on exploitation at latter stages of the run.

3. Hybrid of DE and PSO

DE is one of the most commonly used algorithms in hybrid PSO's. Since DE possesses a strong explorative capability due to its differential mutation, its combination with PSO can significantly enhance swarm diversity and reduce the risk of premature convergence which is the main concern in PSO. Therefore, the salient motivation of using hybrid DE-PSO is to hinder premature convergence. Existent hybrid DE-PSO variants are classified in two sets. First set variants intend to alleviate premature convergence and the second set include variants aiming to set PSO parameters by DE.

3.1 Alternating between DE and PSO

In (Zhang and Xie 2003; Talbi and Batouche 2004), PSO and DE operators are performed alternatively, that is, at odd iterations, the individuals are updated by equations in (3) and at even iterations, DE operator is implemented as follows; For each individual X_i in population, the trial vector X_i' is obtained by

$$\text{If } (rand < CR \text{ or } d = k) \text{ then } X'_{id} = P_{gd} + F(P_{md}(t) - P_{nd}(t)) \quad (5)$$

Where $P_m(t)$ and $P_n(t)$ are the best personal experiences of two randomly selected individuals m and n in population, CR and F are respectively crossover rate and scaling factor of DE which should be tuned experimentally. If X'_i possess a fitness better than X_i, replaces it, otherwise X_i remains unchanged. Incorporation of DE via (5) significantly decreases the number of particles which could become dormant due to the vicinity to P_i and P_g, so the diversity of individuals increases and the risk of premature convergence decreases. Despite this advantage, the additional computational burden required for tuning DE parameters and the sensitivity of hybrid DEPSO with respect to them should be taken into account.

3.2 DE Incorporated in PSO Velocity Update Equations

In (Das, Konar et al. 2005; Das, Abraham et al. 2008; Garcİ-a-Nieto, Alba et al. 2009), in PSO velocity update equation, instead of cognitive term, the particles are perturbed by a differential operator term. That is, for each particle i in the swarm, two distinct particles like k and j are chosen randomly and the difference between their position vector becomes difference vector of DE.

$$\delta_i = X_k - X_j \quad (6)$$

Then new update equation is given by the following equations:

$$V_{id}(t+1) = \begin{cases} \omega V_{id}(t) + F\delta_{id} + C_2 r_2 (P_{gd} - X_{id}) & \text{if } r_d < CR \\ V_{id}(t) & \text{otherwise} \end{cases}$$
(7)

$$X'_i = X_i(t) + V_{id}(t+1)$$

X'_i replaces X_i if its fitness function is better than that of X_i. Addition of this difference term leads to diversification of particles in swarm and reduction in risk of premature convergence.

3.3 Switching between DE and PSO

In (Hao, Guo et al. 2007; Kim and Lee 2009) a hybrid DE-PSO is proposed in which, in each iteration just one of the constituent algorithms operates. The ratio of PSO iterations to DE iterations is called T and can be set by user. In each iteration a random number r_1 in [0,1] is created. If it is less than T, PSO is operated, otherwise DE is implemented as follows (Hao, Guo et al. 2007).

$$X_{id}(t+1) = \begin{cases} P_{id}(t) + F(X_{md} - X_{nd}) & \text{if } r_{2d} < CR \\ X_{id}(t) & \text{otherwise} \end{cases} \quad (8)$$

Where r_{2d} is a random number in [0,1], X_{md} and X_{nd} are two randomly selected individuals in population.

This hybrid technique together with bounce back strategy for maintaining individuals in feasible search space and also reinforcement learning, has led to quality solutions especially in real-world engineering problems.

In another PSO-DE variant (Khamsawang, Wannakarn et al.), for each particle, PSO operates unless either its velocity approaches zero, or it violates search boundaries. Here, The DE with one, two, three and four difference vector is tested on economic dispatch problem and it is found that DE/2 provides more quality solutions for this problem, so, researchers should not confine themselves to applying just DE/1 to hybrid DE/PSO, since DE with other numbers of difference vectors may lead to better results.

In (Yi, Cao et al.; Zhang, Ning et al. 2009), a hybrid DEPSO is put forward wherein DE is implemented in each generation that the difference between individual's current and previous position vectors is defined as their velocity. Furthermore, every K iteration, a combined DE-PSO strategy is implemented.

3.4 Two Population DEPSO

In (Pant, Thangaraj et al. 2009), a hybrid DE-PSO variant is introduced in which, for each particle, only DE is activated unless it fails to improve the fitness value. If it fails, PSO is activated to update that particle. In this variant, it is possible that different individuals work with different heuristics at the same time, some with DE and others with PSO.

In (Niu and Li 2008), a two-population-based scheme is introduced wherein the individuals of one population are enhanced by PSO and the individuals of the other population are evolved by DE. Here, besides DE which diversifies population, there is an inter-population information sharing which decreases the chance of premature convergence. Another two-population based DE-PSO for dealing with constrained problems is introduced in (Liu, Cai et al.).

3.5 Addition of DE once in T iteration

In another hybrid DE-PSO, each individual obeys the PSO update equations, but from time to time, DE is run which may shift individuals to more promising regions and reduce the premature convergence probability (Hendtlass 2001). The ratio of the number of DE runs to the number of PSO runs (T) affects the overall performance of DE-PSO. The more the value of T is, the less is the risk of premature convergence, but the more is computational time. The optimum value of T can be achieved experimentally.

3.6 Bare-bones DE

Bare-bones DE (BBDE), is a hybrid of bare-bones PSO and DE. DE is used to mutate, the attractor associated with each particle, which is defined as a weighted average of its personal and neighborhood best positions as follows (Omran, Engelbrecht et al. 2009).

$$P'_{id}(t) = r_{1d}P_{id}(t) + (1 - r_{1d})P_{gd}(t) \quad (9)$$

And the Position update equations are given by

$$X_{id}(t+1) = \begin{cases} P'_{id}(t) + r_{2d}(X_{md} - X_{nd}) & \text{if } r_{3d} < CR \\ P_{sd}(t) & \text{otherwise} \end{cases} \quad (10)$$

Where m, n, and s are the indices of three randomly selected particles in the swarm, moreover, r_{1d}, r_{2d} and r_{3d} are random numbers in interval [0,1]. According to (10), the region around current personal attractor is searched to find quality solutions. The noticeable point in this variant is that it is parameter-free. Since the scaling factor is a random number, and $CR \sim N(0.5, 0.15)$.

3.7 DEPSO with Prior Crossover

In PSO with prior-crossover DE (PSOPDE) (Xu and Gu 2009), first, PSO is implemented as (11), wherein particles not only are attracted toward personal and global best, but also are attracted toward the average position and average velocity of particles.

$$V_{id}(t+1) = \omega V_{id}(t) + C_1 r_1(P_{id} - X_{id}) + C_2 r_2(P_{gd} - X_{id}) + C_1 r_1(P_{ave,d} - X_{id}) + C_2 r_2(V_{ave,d} - V_{id}) \quad (11)$$

The introduction of these new terms provides particles more information of the evolutionary trend of the whole swarm and enhances search efficiency.

Following PSO, DE starts with two populations. First population consists of target individuals, while the second one records abandoned trial vectors. A prior crossover operation is implemented between the target individual $X_i(t)$ in first population and its corresponding individual in the other population. If the created offspring possesses better fitness value, replaces $X_i(t)$, otherwise the trial vector is generated via traditional mutation and crossover like basic DE. Indeed, in this variant, the PSO new update equations and prior crossover scheme before DE enhance the exploration capability of algorithm, thus provide more accurate and reliable solutions.

3.8 Leader Selector DEPSO

In (Wickramasinghe and Li 2008), DE is used for leader selection in multi-objective PSO. DE via diversification obtains a diverse range of leaders and diminishes the likelihood of getting trapped in local optima.

3.9 Composite PSO

The PSO variants whose parameters are determined by another heuristic are called composite PSO. In (Kannan, Slochanal et al. 2004), DE is used for selecting PSO parameters as follows.

First, positions and velocities of particles (X_i and V_i) and PSO parameters $S_i = [\omega, C_1, C_2]$ are initialised randomly, where i is iteration number. Then for each S_i, X_i and V_i are updated by PSO update equations. Thereafter, DE operators are applied to S_i, to update it. These steps continue till termination criterion is met.

4. Conclusions

Hybridizing PSO with other optimisation techniques can lead to more efficient algorithms, because by hybridization the constituent techniques reinforce each other's strengths and cover each others' shortcomings. DE is one of the algorithms that its hybridization with PSO leads to encouraging outcomes. This paper has presented an analysis on various variants which are hybrids of PSO and DE.

Corresponding Author:
Ahmad Rezaee Jordehi
Department of Electrical Engineering
University Putra Malaysia
UPM Serdang, Selangor 43400, Malaysia
E-mail: ahmadrezaeejordehi@gmail.com

References

1. Kennedy J, Eberhart R. Particle swarm optimization. In, Neural Networks, 1995 Proceedings, IEEE International Conference on; 1995:1942-8 vol.1944.
2. R. Eberhart YS, and J. Kennedy. *Swarm Intelligence*. San Mateo: Morgan Kaufmann; 2001.
3. Shi Y, Eberhart R. A modified particle swarm optimizer. In, Evolutionary Computation Proceedings, 1998 IEEE World Congress on

Computational Intelligence, The 1998 IEEE International Conference on; 1998:69-73.
4. Shi Y, Eberhart RC. Empirical study of particle swarm optimization. In, Evolutionary Computation, 1999 CEC 99 Proceedings of the 1999 Congress on; 1999:1950 Vol. 1953.
5. Zhang WJ, Xie XF. DEPSO: hybrid particle swarm with differential evolution operator. In: IEEE; 2003:3816-21 vol. 3814.
6. Talbi H, Batouche M. Hybrid particle swarm with differential evolution for multimodal image registration. In: IEEE; 2004:1567-72 Vol. 1563.
7. Das S, Abraham A, Konar A. Particle swarm optimization and differential evolution algorithms: technical analysis, applications and hybridization perspectives. Advances of Computational Intelligence in Industrial Systems 2008:1-38.
8. Das S, Konar A, Chakraborty UK. Improving particle swarm optimization with differentially perturbed velocity. In: ACM; 2005:177-84.
9. Garcí-a-Nieto J, Alba E, Apolloni J. Particle swarm hybridized with differential evolution: black box optimization benchmarking for noisy functions. In: ACM; 2009:2343-50.
10. Kim P, Lee J. An integrated method of particle swarm optimization and differential evolution. Journal of mechanical science and technology 2009;23:426-44.
11. Hao ZF, Guo GH, Huang H. A particle swarm optimization algorithm with differential evolution. In: IEEE; 2007:1031-15.
12. Khamsawang S, Wannakarn P, Jiriwibhakorn S. Hybrid PSO-DE for solving the economic dispatch problem with generator constraints. In: IEEE:135-9.
13. Zhang C, Ning J, Lu S, Ouyang D, Ding T. A novel hybrid differential evolution and particle swarm optimization algorithm for unconstrained optimization. Operations Research Letters 2009;37:117-22.
14. Yi W, Cao C, Zhang C. The geometric constraint solving based on hybrid differential evolution and particle swarm optimization algorithm. In: IEEE:692-5.
15. Pant M, Thangaraj R, Abraham A. DE-PSO: a new hybrid meta-heuristic for solving global optimization problems. New Mathematics and Natural Computation, Accepted for publication 2009.
16. Niu B, Li L. A novel PSO-DE-based hybrid algorithm for global optimization. Advanced Intelligent Computing Theories and Applications With Aspects of Artificial Intelligence 2008:156-63.
17. Liu H, Cai Z, Wang Y. Hybridizing particle swarm optimization with differential evolution for constrained numerical and engineering optimization. Applied Soft Computing;10:629-40.
18. Hendtlass T. A combined swarm differential evolution algorithm for optimization problems. Engineering of Intelligent Systems 2001:11-18.
19. Omran MGH, Engelbrecht AP, Salman A. Bare bones differential evolution. European Journal of Operational Research 2009;196:128-39.
20. Xu W, Gu X. A hybrid particle swarm optimization approach with prior crossover differential evolution. In: ACM; 2009:671-8.
21. Wickramasinghe U, Li X. Choosing leaders for multi-objective PSO algorithms using differential evolution. Simulated Evolution and Learning 2008:249-58.
22. Kannan S, Slochanal S, Subbaraj P, Padhy NP. Application of particle swarm optimization technique and its variants to generation expansion planning problem. Electric Power Systems Research 2004;70:203-10.

9/25/2018