An Analysis to Hybrids of Particle Swarm Optimisation and Genetic Algorithms

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 and most efficient optimisation techniques that exposes desirable computational behavior. However, hybridizing it with other optimisation techniques can lead to even 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 genetic algorithm. This paper presents a comprehensive analysis on various variants which are hybrids of PSO and GA operators.

[Ahmad Rezaee Jordehi, Jasronita Jasni. **An Analysis to Hybrids of Particle Swarm Optimisation and Genetic Algorithms.** *Biomedicine and Nursing* 2018;4(2): 79-83]. ISSN 2379-8211 (print); ISSN 2379-8203 (online). http://www.nbmedicine.org. 14. doi:10.7537/marsbnj040218.14.

Keywords: Particle swarm optimisation, Genetic algorithms, 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 in 1995 (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.
- It is efficient in tackling multi-objectives, multi-modalities, constraints, discrete/integer variables.

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 a comprehensive analysis on various variants which are hybrids of PSO and GA 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-GA variants is provided. Finally, drawing conclusions and proposing some directions for future research in this area is implemented 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 (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. Hybrids of GA and PSO

GA is the most commonly used algorithm in hybrid PSO’s and there exist a plethora of hybrid GA-PSO schemes in literature. All these schemes have been categorised and explained as follows.

3.1 PSO with Selection Operator

Selection operator’s functionality is to redirect search effort towards those positions in search space that have proved to be more promising, thus it enhances exploitive capability. In PSO, although there is a form of implicit selection mechanism, but it is too weak. In (Naka, Genji et al. 2003) and (Jong-Bae, Yun-Won et al.), an explicit selection operator has been incorporated in PSO. For implementing it, a prespecified number of particles are ranked according to their fitnesses. Then, the positions and velocities of the half fitter particles replace the positions and particles of the half less fit particles while their personal best does not change. These replacements are done before updating velocities and positions of particles. Selection mechanism enhances exploitive capability of PSO, so it generally can conduct better fine tuning search and achieve higher quality solutions. The results show that this hybrid PSO outperforms basic PSO on three of four test functions used. In these functions, hybrid PSO lags basic PSO in first iterations, but surpasses it thereafter and achieves more accurate final solutions. On the other test function, Hybrid PSO's performance is significantly worse than basic PSO which again approves "no free lunch" theorem. Since selection is expected to attenuate exploration capability of PSO, it seems necessary to test it on complex multimodal problems to probe its capability for avoiding local optima.

3.2 PSO with Crossover Operator

In (Jong-Bae, Yun-Won et al.), crossover operator is adopted in order to increase swarm diversity and prevent premature convergence. In this variant, after calculating $X_i(t+1)$ via equations in (4), $X'_{id}(t+1)$ is calculated by:

$$X'_{id}(t+1) = \begin{cases} X_{id}(t+1) & r_{id} \le CR \\ P_{i,d}(t) & \text{otherwise} \end{cases} \quad (5)$$

Where CR is crossover rate and r_{id} is random number in interval [0,1] and $P_{i,d}(t)$ is dth dimension of the best visited position of particle i till iteration k.

Equation (5) implies that some dimensions of new particle's position are replaced by corresponding elements of its personal best according to crossover rate. In (Jong-Bae, Yun-Won et al.), crossover in companion with a chaotic inertia weight strategy is applied to economic dispatch problem in power systems. Tests on large-scale Korean power system show the efficiency of this variant in terms such as global optimality and accuracy. The main criticism can be cast on these variants is the additional computational effort needed for experimentally tuning crossover rate.

In (Yuexin and Honggeng), a crossover operator relatively similar to (Naka, Genji et al. 2003) is adopted.

$$X'_{id}(t+1) = \begin{cases} X_{id}(t+1) & r_{id} \le CR \\ P_{g,d}(t) & \text{otherwise} \end{cases} \quad (6)$$

That is, some dimensions of new particle's position are replaced by corresponding elements of swarm's best according to crossover rate.

In (Yin 2006), this type of crossover is introduced:

$$X_{id}(t+1) = \omega(0,\omega_1)X_{id}(t) + \omega(\omega_1,\omega_2)P_{i,d}(t) + \omega(\omega_2,1)P_{g,d}(t) \quad (7)$$

Where $\omega(a,b) = \begin{cases} 1 & a \le r_1 < b \\ 0 & \text{otherwise} \end{cases}$

Inspired by (Yin 2006), a modified heuristic crossover invoked by differential evolution is applied to PSO as follows (Chen, Zhao et al. 2009).

$$X_{id}(t+1) = \begin{cases} X_{id}(t) + P_r\left(P_{g,d}(t) - P_{i,d}(t)\right) & r_{id} \le \omega_1 \\ P_{i,d}(t) + P_r\left(P_{g,d}(t) - X_{id}(t)\right) & r_{id} \le \omega_2 \\ P_{g,d}(t) + P_r\left(P_{i,d}(t) - X_{id}(t)\right) & \text{otherwise} \end{cases} \quad (8)$$

Where P_r , ω_1 and ω_2 are user-defined parameters which whose optimum values can be determined experimentally.

Also in (Chen, Mimori et al. 2009) and (Shunmugalatha and Slochanal 2008), the concepts of crossover and subpopulation are incorporated into PSO to hinder premature convergence.

3.3 PSO with Mutation Operator

Due to the high probability of occurring premature convergence in PSO, in most of PSO variants, especially in variants adapted for multimodal problems, mutation operator as one of salient genetic algorithm operators is adopted (Esmin, Lambert-Torres et al. 2006). It significantly diminishes the chance of premature convergence by increasing swarm diversity.

3.4 PSO with all GA Operators

In some variants, genetic algorithm with its all operators is hybridised with PSO (Ziari, Ledwich et al.; Juang 2004; Ru and Jianhua 2008). In (Ziari, Ledwich et al.), the individuals are started as PSO particles, then after updating positions, velocities, P_{best}s, and g_{best}, in order to increase swarm diversity and prevent premature convergence, half of particles are affected by GA operators, while the other half remains unaffected by GA.

In another variant (Juang 2004), GA and PSO operate on the same population. In each generation, individuals for next generation are created by enhancement and crossover operators as follows; by calculating the fitness of all individuals and ranking them, the top half best-performing particles are recognised and called as "elites". But instead of GA in which elites move to next generation directly, here, at first, they are enhanced via constituting a swarm and sharing information among themselves, inspired by "maturing" phenomenon in nature. Indeed, they enhance themselves according to equation (3), then enhanced form of elites go to next generation and occupy half of the generation. For filling the second half of next generation, tournament selection selects parents from the elites and crossover is applied to selected elites. The crossover breeds offsprings which constitute the second half of next generation. The application of this variant on training recurrent neural networks demonstrate its superiority over pure GA and pure PSO.

3.5 Hybrids with switching between PSO and GA

In a few hybrid variants, some iterations of run are implemented by PSO and some others by GA (Robinson, Sinton et al. 2002; Mohammadi and Jazaeri 2007; Yang, Chen et al. 2007).

In (Robinson, Sinton et al. 2002), the population of one of the algorithms is taken as the starting population for the next algorithm when the improvement begins to level off. Both the GA-PSO (first GA then PSO) and PSO-GA (first PSO then GA) have been tested on test functions and the results show outperformance of PSO-GA over GA-PSO, pure PSO and pure GA in most of test functions. The outperformance of PSO-GA is as per expectation, since according to (Angeline 1998), PSO can find reasonable quality solutions so faster than other evolutionary algorithms, but when the swarm is going to be in equilibrium, the evolution process is stagnated, and PSO does not provide an efficient fine tuning, thus by using PSO-GA in a sequential manner, the strengths of both algorithms are extracted. Also in (Mohammadi and Jazaeri 2007) and (Yang, Chen et al. 2007) hybrid algorithms start with PSO and switch to GA after a prespecified number of iterations.

3.6 Hybrids with Concurrent Existence of both PSO and GA Populations

In (Krink and Lأ¸vbjerg 2002), a life-cycle model is put forward which is inspired by the ability of individuals in nature to actively decide about their kind of lifestyle in response to their success in current environment. In this model, each individual according to recent search progress can decide whether to join population of GA, swarm of PSO or become a solitary stochastic hill climber, that is, population of all three algorithms work simultaneously, though their population size varies during the course of the run. The model starts with PSO particles, and can turn into GA individuals, or hill climbers during the run. The stage for individuals is switched when no fitness improvement is found after a prespecified number of iterations.

In (Premalatha and Natarajan 2009), the total number of iterations is halved between GA and PSO. In first half of iterations, GA individuals are run while in second half, PSO swarm is run, initialised with solutions of GA. This mechanism in addition to strategies for making perturbation in P_{best}s, and g_{best}, during stagnation have resulted in promising results on some test functions.

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. GA 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 GA operators.

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; 1995:1942-1948 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. Naka S, Genji T, Yura T, Fukuyama Y. A hybrid particle swarm optimization for distribution state estimation. Power Systems, IEEE Transactions on 2003;18:60-68.
6. Jong-Bae P, Yun-Won J, Joong-Rin S, Lee KY. An Improved Particle Swarm Optimization for Nonconvex Economic Dispatch Problems. Power Systems, IEEE Transactions on;25:156-166.
7. Yuexin D, Honggeng Y. A New Approach for Reactive Power/Voltage Optimization Control of Regional Grid. In, Power and Energy Engineering Conference (APPEEC), 2010 Asia-Pacific:1-5.
8. Yin PY. Genetic particle swarm optimization for polygonal approximation of digital curves. Pattern Recognition and Image Analysis 2006;16:223-233.
9. Chen P, Zhao C, Li J, Liu Z. Solving the Economic Dispatch in Power System via a

Modified Genetic Particle Swarm Optimization. In: IEEE; 2009:201-204.
10. Chen YW, Mimori A, Lin CL. Hybrid particle swarm optimization for 3-D image registration. In: IEEE; 2009:1753-1756.
11. Shunmugalatha A, Slochanal S. Optimum cost of generation for maximum loadability limit of power system using hybrid particle swarm optimization. International Journal of Electrical Power & Energy Systems 2008;30:486-490.
12. Esmin A, Lambert-Torres G, Alvarenga GB. Hybrid evolutionary algorithm based on PSO and GA mutation. In: IEEE; 2006:57-57.
13. Ziari I, Ledwich G, Ghosh A. Optimal voltage support mechanism in distribution networks. Generation, Transmission & Distribution, IET;5:127-135.
14. Juang CF. A hybrid of genetic algorithm and particle swarm optimization for recurrent network design. Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on 2004;34:997-1006.
15. Ru N, Jianhua Y. A GA and particle swarm optimization based hybrid algorithm. In: IEEE; 2008:1047-1050.
16. Robinson J, Sinton S, Rahmat-Samii Y. Particle swarm, genetic algorithm, and their hybrids: optimization of a profiled corrugated horn antenna. In: IEEE; 2002:314-317 vol. 311.
17. Mohammadi A, Jazaeri M. A hybrid particle swarm optimization-genetic algorithm for optimal location of SVC devices in power system planning. In: IEEE; 2007:1175-1181.
18. Yang B, Chen Y, Zhao Z. A hybrid evolutionary algorithm by combination of PSO and GA for unconstrained and constrained optimization problems. In: IEEE; 2007:166-170.
19. Angeline P. Evolutionary optimization versus particle swarm optimization: Philosophy and performance differences. In: Springer; 1998:601-610.
20. Krink T, Lí¸vbjerg M. The lifecycle model: combining particle swarm optimisation, genetic algorithms and hillclimbers. Parallel Problem Solving from Natureâ€”PPSN VII 2002:621-630.
21. Premalatha K, Natarajan A. Hybrid PSO and GA for global maximization. Int J Open Problems Compt Math 2009;2:597-608.

6/25/2018