Websites:
http://www.jofamericanscience.org
http://www.sciencepub.net

Emails:
editor@sciencepub.net
sciencepub@gmail.com

Journal of American Science

MARSLAND PRESS
Multidisciplinary Academic Journal Publisher

Particle Swarm Optimisation in Dynamic Environments; Challenges and Strategies

Ahmad Rezaee Jordehi, Jasronita Jasni

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

Abstract: A host of practical optimisation problems are dynamic. That is, their objective function and/or constraints change over time. These problems are so formidable. Among various optimisation techniques, particle swarm optimisation (PSO) has proved to be promising in solving dynamic problems. In this paper, the challenges existent in dynamic environments are explained and PSO strategies for tackling dynamic problems are analysed.
[Jordehi R, Jasni J. **Particle Swarm Optimisation in Dynamic Optimisation Problems; Challenges and Strategies,** *J Am Sci* 2024;20(2):31-36]. ISSN 1545-1003 (print); ISSN 2375-7264 (online). http://www.jofamericanscience.org. 05. doi:10.7537/marsjas200224.05.

Keywords: Particle swarm optimisation, Dynamic optimisation problem.

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 optimisation technique introduced by Kennedy and Eberhart (Kennedy & Eberhart, 1995). It belongs to the family of swarm intelligence computational techniques and is inspired of social interaction in human beings and animals.

Some PSO aspects that make it potent 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.
- Roughly, initial solutions do not affect its computational behavior.
- Its behavior is not highly affected by increase in dimensionality.

However, basic PSO variants are merely applicable to static optimisation problems while many real-world optimization problems are dynamic. Therefore, for solving dynamic problems, typical PSO variants should be modified. In this paper the challenges of dynamic environments are introduced and various PSO variants specially designed for dynamic problems are analysed. The paper is organised as follows; in section 2, an overview of PSO is presented. In section 3, existent PSO variants are analysed. Finally, drawing conclusions and proposing some directions for future research in this area is presented in section 4.

2. Basic Concepts and Variants of PSO

PSO is launched with the random initialisation of a swarm of 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 cognitive and social acceleration coefficients respectively and r_{1d} and r_{2d} are two random numbers with uniform distribution in the interval [0,1].

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, Shi, & Kennedy, 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 & Eberhart, 1998; Shi & 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. PSO for Dynamic Environments

Many real-world optimization problems are dynamic, that is, their objective function and/or constraints vary over time. The general representation of a dynamic optimization problem (DOP) is:

$$\text{Minimize } f(X,t)$$

$$\text{Subject to } g(X,t) \le 0$$

In DOPs due to the change of objective functions or constraints, the position and value of optima varies over time. The role of optimization technique is to track the changing optimum/optima. If the change in is radical, the best option is to implement optimization process from scratch (Branke, 2002). However, in most practical instances, the changes are gradual. If this is the case, it is possible to speed up optimization after an environmental change via utilising some of the information gathered during the optimisation so far.

There are two main difficulties that should be addressed in DOPs:

1. Outdated memory: When the environment changes, previously good solutions may no longer be good and mislead the swarm towards false optima. This issue is more acute when the environment change is severe.
2. Diversity loss: If the environment changes when the new optimum is within the collapsing swarm, it is expected that new optima can be traced successfully and promptly. However if the new optimum is outside the swarm's expansion, particles' low velocities inhibit re-diversification and the swarm may even oscillate around a false attractor in a phenomenon called "linear collapse." So, the new optimum is difficult to be traced.

3.1 Addressing Challenges in DOP's

For addressing the challenges in dynamic environments, three following crucial actions should be undertaken:

3.1.1 Detecting Change

In many applications, the time of change occurrence is known to the system. Otherwise, it should be detected. Change detection strategy should efficiently detect the change and trigger "response mechanism." In (Richter, 2009b), two major types of change detection mechanisms, that is, population-based and sensor-based mechanisms have been introduced. In population-based approach, the fitness evaluations of the population is used, while in the latter approach, additional measurement of the landscape's fitness on prescribed points is utilised.

In population-based approach, the sets $S(t)$ and $S(t+1)$ are formed consisting of fitness functions of individuals at iterations t and $t+1$ respectively. Then the change detection problem is transformed into the problem of testing whether the data set $S(t)$ and $S(t+1)$ are coming from different distributions or not. For judging about this, a statistical hypothesis testing method is applied.

On the other hand, sensor-based approach is based on implementing additional measurements in the fitness landscape using "fitness landscape sensors." If any of the sensors detects an altered fitness value, the change is considered to be occurred. The advantage of this approach in comparison to population-based approach is that it does not need elaborated statistical analysis and there can be no false positives. However, additional fitness evaluation is required.

In (Richter, 2009a), artificial immune system is applied for change detection. A negative selection algorithm has been used to decide on whether or not the fitness landscape has changed. This is solely done with fitness information from the population on a sample base. Also in (Richter & Dietel), a change detection strategy for constrained environments has been put forward.

Despite all above-mentioned efficient change detection strategies, in dynamic PSO literature, change detection is usually done by re-evaluating one or more personal bests, and if the fitness value of at least one of them has changed, the change occurrence is concluded. In this paper, during the explanation of various PSO variants for dynamic environments, the change detection schemes adopted in them will be also explained.

3.1.2 Memory Update

For solving "outdated memory" problem, commonly either re-evaluating or forgetting the memory is used. In the latter, each particle's memory is set to the particle's current position and the global best is updated such that $P_g = \arg\min f(P_i)$.

3.1.3 Diversity Enhancement

Diversity loss is considered as the most challenging issue in DOPs. For solving it, either a diversity-enhancement mechanism should be invoked at change (or at pre-determined intervals), or sufficient diversity has to be ensured at all times. There are four principle mechanisms for diversity-enhancement: re-diversification, repulsion, multi-populations and dynamic neighborhood topology.

3.2 Common Benchmarks and Metrics in DOP Literature

Now, before explaining various PSO variants for DOP's, benchmarks and metrics that are commonly used in specialised DOP literature, are introduced.

3.2.1 Moving Peaks Benchmark (MPB)

These benchmarks that somehow represent real-world dynamic optimization problems are commonly used in the specialised literature. In MPB problems, the optima can vary in three terms; the height, width and location of peaks. MPB is defined as (Branke, 1999):

$$F(X,t) = \operatorname{Max} \frac{H_i(t)}{1 + W_i(t)\sum_{d=1}^{n}\left(X_d(t) - X_{id}(t)\right)^2} \quad \text{for } i = 1,\ldots,p \qquad (5)$$

Where $H_i(t)$ and $W_i(t)$ are the height and width of peak i at time t, respectively. $X_{id}(t)$ represents the dth dimension of position vector of peak i at time t and p denotes the number of peaks. The position of each peak is shifted in a random direction as proceeds:

$$X_i(t) = X_i(t-1) + V_i(t)$$

$$V_i(t) = \frac{s}{|r + V_i(t-1)|}\left((1-\lambda)r + \lambda V_i(t-1)\right) \qquad (6)$$

Where $V_i(t)$ is called shift vector and is a linear combination of a random vector r and the previous shift vector $V_i(t-1)$. □ is called correlation factor and indicates the level of correlation between two successive environmental changes. The final parameter is U and represents that the environment change happens every U function evaluations.

3.2.2 Performance Metric

For gauging the performance of a dynamic optimization technique, commonly a metric called "offline error" is used which is defined via:

$$e = \frac{1}{K}\sum_{K=1}^{K}(h_K - f_K) \qquad (7)$$

Where f_K is the best solution obtained just before the kth environmental change and h_K is optimum value of the k th environment. K is the total number of environmental changes.

3.3 Classification of PSO Variants Adapted for DOP's

Here, due to the importance of diversity-enhancement strategies, all different variants for DOPs are classified according their diversity-enhancement schemes. According to this criterion, most of them are classified into following main groups.

3.3.1 Repulsion-based Variants

A constant degree of swarm diversity can be maintained at all times through some type of repulsive strategies. One repulsive strategy is to use charged particles in swarm wherein diversity is maintained by coulomb repulsion among particles. In one type, called charged PSO, all swarm particles are charged while in second type, called atomic PSO, just some particles are charged and others are neutral particles. Charged particles through diversity enhancement help PSO to trace new optimum (Blackwell, 2003; Blackwell & Branke, 2004; T. M. Blackwell & P. Bentley, 2002; T. M. Blackwell & P. J. Bentley, 2002). In (T. M. Blackwell & P. J. Bentley, 2002), the performance of conventional PSO, charged PSO and atomic PSO in tackling with DOP's are evaluated and experiments have proved the superiority of atomic PSO over conventional and charged PSO's while in (Blackwell, 2003), dynamic environments, by categorising dynamic problems into three types according to their severity, for each type one variant among conventional, charged and atomic PSO is recommended.

In (Blackwell & Branke, 2004) and (Blackwell & Branke, 2006; Sun, Lai, Xu, & Chai, 2007; Sun, Xu, & Fang, 2006; Zhao, Sun, Chen, & Xu, 2009), the charged particle idea has been simplified to the quantum particle. Quantum particles are re-positioned in each iteration within a hypersphere of r_{cloud} centered on the P_g according to a specified probability distribution, that is quantum particles do not move according to PSO's regular update equations. In (Sun, et al., 2007), the center of position distribution is the personal best of a randomly selected particle instead of P_g and in (Sun, et al., 2006), a mutation operator is exerted on g_{best} to enhance diversity further. In literature, various probability distributions including Gaussian, uniform volume and non-uniform volume are used for quantum particles. In comparison with charged particles, quantum particles behave better thanks to lower complexity and easy controllability.

3.3.2 Re-randomisation-based Variants

In (Hu & Eberhart, 2002), for responding to the change, re-randomisation of part or whole of the swarm, and re-randomisation of g_{best} and resetting the whole particles are tested. The experiments show that lower re-randomization rates outperform higher re-randomization rates when change severity is low and there is no significant difference when the change severity is high.

For change detection, in contrary to "changed-g_{best} value" method in (Hu & Eberhart, 2001), "fixed-g_{best} value" method is used wherein the g_{best} value and the second g_{best} value are monitored. If they do not change during a certain number of iterations, it is concluded that the change has occurred.

Two drawbacks of this variant are that the algorithm is strongly sensitive to the number of iterations which g_{best} should be fixed and the technique just has been tested on some simple test functions.

3.3.3 Dynamic Neighborhood Topology-based Variants

In (Janson & Middendorf, 2004) and (Janson & Middendorf, 2006), hierarchical PSO and also its extension called partitioned hierarchical PSO (PH-PSO) have been applied to the DOP's in the hope that these variants due to their dynamic neighborhood topology would be more compatible with dynamic environments. In PH-PSO, after change detection which is done by changed-g_{best} value method, the hierarchy is partitioned into a set of sub-hierarchies or sub-swarms. These sub-swarms continue to search for the optimum independently. After a certain number of iterations, the subswarms are re-unified by connections of the hierarchy that have been cut. In a more extended variant called adaptive PH-PSO, the number of these iterations is adapted. In (Janson & Middendorf, 2006), a new change detection strategy called "hierarchy monitoring strategy" is proposed. In this strategy, at each iteration, the total number of swaps is measured and if it exceeds a certain threshold, the change is considered to have occurred. The rationale behind this is that when change occurs, the number of particles swapped between nodes of tree drastically increases. The advantage of this strategy is that it does not need additional function evaluations. The experiments reveal that PH-PSO outperforms H-PSO and conventional PSO.

Nevertheless, besides the above-mentioned dynamic neighborhood topologies, some static topologies have also been utilised for tackling DOP's. In (Xiaodong & Hoa, 2003), in a variant named fine-grained PSO, Von-Neumann neighborhood topology is adopted wherein each particle is neighboured to the four particles in its four sides. This neighborhood topology provides slower information propagation in comparison to g_{best} PSO, leading to higher diversity level that prompts superior performance in dynamic environments. The experiments strongly approve its superiority. Furthermore, in (Zheng & Liu, 2009), the swarm is partitioned into two subswarms; G-subswarm with a g_{best} neighborhood topology responsible for finding new optima and L-subswarm with a l_{best} neighborhood topology responsible for compensating the diversity loss. The two subswarms exchange their best particles in some checkpoints. Also a mutation operator is applied to g_{best} to ensure enough diversity.

3.3.4 Multi-Swarm Variants

In these variants, the swarm is partitioned into a number of subswarms with the aim to position each subswarm on a different promising peak of the landscape. Here, in addition to multi-swarm concept, three following diversity operators are adopted (Blackwell & Branke, 2006; Blum, Merkle, Blackwell, Branke, & Li, 2008).

1. Quantum or charged particles: Diversity loss incurred by environmental change is counterbalanced by using charged or quantum particles. In charged multi-swarm PSO, consisting of neutral and charged particles, diversity is maintained by coulomb repulsion among charged particles while in Quantum multi-swarm PSO, consisting neutral and quantum particles whereas the quantum particles are randomised within a ball of radius r_{cloud} centered on the subswarm attractor.
2. Anti-convergence: When the number of subswarms is less than the number of peaks, some peaks are not covered and if those peaks become the optima after environmental change, cannot be easily tracked. Therefore, whenever all subswarms have converged, that is, when their neutral swarm size gets less than convergence radius, r_{conv}, anti-convergence operator expels the worst sub-swarm from its peak and reinitialises it in the search space. Therefore, at each time, there is at least one subswarm patrolling search space for new peaks.
3. Exclusion: When the attractor of two subswarms are within an exclusion radius, r_{excl}, the subswarm with worse objective function is expelled and reinitialised in the search space. Indeed exclusion is a local interaction between colliding swarms, preventing them from settling on the same peak, or in words, it maintains inter-subswarm diversity.

The experiments on benchmark functions show that the above-mentioned multi-swarm PSO behaves well but its main issue is the large number of the parameters that are difficult to be set. The number of sub-swarms, the number of quantum particles, quantum cloud radius (or charge in charged multi-swarms), exclusion distance and convergence radius are the parameters that should be tuned.

4. Conclusions and future research directions

PSO has demonstrated desirable performance in solving dynamic optimisation problems. In this paper PSO variants devised for dynamic problems are analysed. Based on conducted analysis, the followings are proposed as some directions for future research in this area.

- ❖ Applying other diversity enhancement mechanisms such as various mutation forms, self-organised criticality, prey and predator scheme, comprehensive learning scheme and dissipative strategy may lead to more superior outcomes in dynamic environments and is recommended.
- ❖ While theoretical analysis on PSO in dynamic environments may lead to deeper understanding in this area, it has not been conducted yet and is recommended for future research.

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. Neural Networks, 1995. Proceedings., IEEE International Conference on, Vol. 4 (1995) 1942-1948.
[2]. Eberhart, R.C., Shi, Y., Kennedy, J.: Swarm intelligence. Elsevier (2001)
[3]. Shi, Y., Eberhart, R.: A modified particle swarm optimizer. Evolutionary Computation Proceedings, 1998. IEEE World Congress on Computational Intelligence., The 1998 IEEE International Conference on (1998) 69-73
[4]. Shi, Y., Eberhart, R.C.: Empirical study of particle swarm optimization. Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on, Vol. 3 (1999) 1950 Vol. 1953
[5]. Branke, J.: Evolutionary Optimization in Dynamic Environments. (2002)
[6]. Richter, H.: Detecting change in dynamic fitness landscapes. IEEE (2009) 1613-1620
[7]. Richter, H.: Change detection in dynamic fitness landscapes: An immunological approach. IEEE (2009) 719-724
[8]. Richter, H., Dietel, F.: Change detection in dynamic fitness landscapes with time-dependent constraints. IEEE 580-585
[9]. Branke, J.: The moving peaks benchmark. URL: http://www. aifb. uni-karlsruhe. de/âˆ¼ jbr/MovPeaks/movpeaks (1999)
[10]. Blackwell, T.M., Bentley, P.: Don't push me! Collision-avoiding swarms. Vol. 2. IEEE (2002) 1691-1696

[11]. Blackwell, T.M., Bentley, P.J.: Dynamic search with charged swarms. Citeseer (2002) 19-26
[12]. Blackwell, T., Branke, J.: Multi-swarm optimization in dynamic environments. Applications of Evolutionary Computing (2004) 489-500
[13]. Blackwell, T.: Swarms in dynamic environments. Springer (2003) 200-200
[14]. Blackwell, T., Branke, J.: Multiswarms, exclusion, and anti-convergence in dynamic environments. Evolutionary Computation, IEEE Transactions on **10** (2006) 459-472
[15]. Zhao, J., Sun, J., Chen, W., Xu, W.: Tracking Extrema in Dynamic Environments with Quantum-behaved Particle Swarm Optimization. Vol. 2. IEEE (2009) 103-108
[16]. Sun, J., Lai, C., Xu, W., Chai, Z.: A novel and more efficient search strategy of quantum-behaved particle swarm optimization. Adaptive and Natural Computing Algorithms (2007) 394-403
[17]. Sun, J., Xu, W., Fang, W.: A diversity-guided quantum-behaved particle swarm optimization algorithm. Simulated Evolution and Learning (2006) 497-504
[18]. Hu, X., Eberhart, R.C.: Adaptive particle swarm optimization: detection and response to dynamic systems. Vol. 2. IEEE (2002) 1666-1670
[19]. Hu, X., Eberhart, R.: Tracking dynamic systems with PSO: where's the cheese. (2001) 80-83
[20]. Janson, S., Middendorf, M.: A hierarchical particle swarm optimizer for dynamic optimization problems. Applications of Evolutionary Computing (2004) 513-524
[21]. Janson, S., Middendorf, M.: A hierarchical particle swarm optimizer for noisy and dynamic environments. Genetic Programming and Evolvable Machines **7** (2006) 329-354
[22]. Xiaodong, L., Khanh Hoa, D.: Comparing particle swarms for tracking extrema in dynamic environments. Evolutionary Computation, 2003. CEC '03. The 2003 Congress on, Vol. 3 (2003) 1772-1779 Vol.1773
[23]. Zheng, X., Liu, H.: A different topology multi-swarm PSO in dynamic environment. IT in Medicine & Education, 2009. ITIME '09. IEEE International Symposium on, Vol. 1 (2009) 790-795
[24]. Blum, C., Merkle, D., Blackwell, T., Branke, J.r., Li, X.: Particle Swarms for Dynamic Optimization Problems. Swarm Intelligence. Springer Berlin Heidelberg (2008) 193-217
[25]. Blackwell, T.: Particle swarm optimization in dynamic environments. Evolutionary Computation in Dynamic and Uncertain Environments (2007) 29-49

2/21/2024