Swarms in Dynamic Environments

T.M. Blackwell

Department of Computer Science, University College London, Gower Street, London, UK
tim.blackwell@ieee.org

Abstract. Charged particle swarm optimization (CPSO) is well suited to the dynamic search problem since inter-particle repulsion maintains population diversity and good tracking can be achieved with a simple algorithm. This work extends the application of CPSO to the dynamic problem by considering a bi-modal parabolic environment of high spatial and temporal severity. Two types of charged swarms and an adapted neutral swarm are compared for a number of different dynamic environments which include extreme 'needle-in-the-haystack' cases. The results suggest that charged swarms perform best in the extreme cases, but neutral swarms are better optimizers in milder environments.

1 Introduction

Particle Swarm Optimization (PSO) is a population based optimization technique inspired by models of swarm and flock behavior [1]. Although PSO has much in common with evolutionary algorithms, it differs from other approaches by the inclusion of a solution (or particle) velocity. New potentially good solutions are generated by adding the velocity to the particle position. Particles are connected both temporally and spatially to other particles in the population (swarm) by two accelerations. These accelerations are spring-like: each particle is attracted to its previous best position, and to the global best position attained by the swarm, where 'best' is quantified by the value of a state function at that position. These swarms have proven to be very successful in finding global optima in various static contexts such as the optimization of certain benchmark functions [2].

The real world is rarely static, however, and many systems will require frequent re-optimization due to a dynamic environment. If the environment changes slowly in comparison to the computational time needed for optimization (i.e. to within a given error tolerance), then it may be hoped that the system can successfully re-optimize. In general, though, the environment may change on any time-scale (temporal severity), and the optimum position may change by any amount (spatial severity). In particular, the optimum solution may change discontinuously, and by a large amount, even if the dynamics are continuous [3]. Any optimization algorithm must therefore be able to both detect and respond to change.

E. Cantú-Paz et al. (Eds.): GECCO 2003, LNCS 2723, pp. 1–12, 2003.

Recently, evolutionary techniques have been applied to the dynamic problem [4, 5, 6]. The application of PSO techniques is a new area and results for environments of low spatial severity are encouraging [7, 8]. CPSO, which is an extension of PSO, has also been applied to more demanding environments, and found to outperform the conventional PSO [9, 10]. However, PSO can be improved or adapted by incorporating change detecting mechanisms [11]. In this paper we compare adaptive PSO with CPSO for various dynamic environments, some of which are severe both spatially and temporally. In order to do this, we use a model which enables simple testing for the three types of dynamism defined by Eberhart, Shi and Hu [7, 11].

2 Background

The problem of optimization within a general and unknown dynamic environment can be approached by a classification of the nature of the environment and a quantification of the difficulty of the problem. Eberhart, Shi and Hu [7, 11] have defined three types of dynamic environment. In type I environments, the optimum position $\boldsymbol{x}_{opt}$, defined with respect to a state function f, is subject to change. In type II environments, the value of f at $\boldsymbol{x}_{opt}$ varies and, in type III environments, both $\boldsymbol{x}_{opt}$ and $f(\boldsymbol{x}_{opt})$ may change. These changes may occur at any time, or they may occur at regular periods, corresponding, for example, to a periodic sensing of the environment. Type I problems have been quantified with a severity parameter s, which measures the jump in optimum location.

Previous work on PSO in dynamic environments has focused on periodic type I environments of small spatial severity. In these mild environments, the optimum position changes by an amount $s\boldsymbol{I}$, where $\boldsymbol{I}$ is the unit vector in the n-dimensional search space of the problem. Here, 'small' is defined by comparison with the dynamic range of the internal variables $\boldsymbol{x}$. Comparisons of CPSO and PSO have also been made for severe type I environments, where s is of the order of the dynamic range [9]. In this work, it was observed that the conventional PSO algorithm has difficulty adjusting in spatially severe environments due to over specialization. However, the PSO can be adapted by incorporating a change detection and response algorithm [11].

A different extension of PSO, which solves the problem of change detection and response, has been suggested by Blackwell and Bentley [10]. In this extension (CPSO), some or all of the particles have, in analogy with electrostatics, a 'charge'. A third collision-avoiding acceleration is added to the particle dynamics, by incorporating electrostatic repulsion between charged particles. This repulsion maintains population diversity, enabling the swarm to automatically detect and respond to change, yet does not diminish greatly the quality of solution. In particular, it works well in certain spatially severe environments [9].

Three types of particle swarm can be defined: neutral, atomic and fully-charged. The neutral swarm has no charged particles and is identical with the conventional PSO. Typically, in PSO, there is a progressive collapse of the swarm towards the best position, with each particle moving with diminishing amplitude around the best position. This ensures good exploitation, but diversity is lost. However, in a swarm of 'charged' particles, there is an additional collision avoiding acceleration. Animations for this swarm reveal that the swarm maintains an extended shape, with the swarm centre close to the optimum location [9, 10]. This is due to the repulsion which works against complete collapse. The diversity of this swarm is high, and response to environment change is quick. In an 'atomic' swarm, 50% of the particles are charged and 50% are neutral. Animations show that the charged particles orbit a collapsing nucleus of neutral particles, in a picture reminiscent of an atom. This type of swarm therefore balances exploration with exploitation. Blackwell and Bentley have compared neutral, fully charged and atomic swarms for a type-I time-dependent dynamic problem of high spatial severity [9]. No change detection mechanism is built into the algorithm. The atomic swarm performed best, with an average best values of f some six orders of magnitude less than the worst performer (the neutral swarm).

One problem with adaptive PSO [11], is the arbitrary nature of the algorithm (there are two detection methods and eight responses) which means that specification to a general dynamic environment is difficult. Swarms with charge do not need any adaptive mechanisms since they automatically maintain diversity. The purpose of this paper is to test charged swarms against a variety of environments, to see if they are indeed generally applicable without modification.

In the following experiments we extend the results obtained above by considering time-independent problems that are both spatially and *temporally* severe. A model of a general dynamic environment is introduced in the next section. Then, in section 4, we define the CPSO algorithm. The paper continues with sections on experimental design, results and analysis. The results are collecting together in a concluding section.

3 The General Dynamic Search Problem

The dynamic search problem is to find $\boldsymbol{x}_{opt}$ for a state function $f(\boldsymbol{x}, \boldsymbol{u}(t))$ so that $f(\boldsymbol{x}_{opt}, t) \equiv f_{opt}$ is the instantaneous global minimum of f. The state variables are denoted $\boldsymbol{x}$ and the influence of the environment is through a (small) number of control variables $\boldsymbol{u}$ which may vary in time. No assumptions are made about the continuity of $\boldsymbol{u}(t)$, but note that even smooth changes in $\boldsymbol{u}$ can lead to discontinuous change in $\boldsymbol{x}_{opt}$. (In practice a sufficient requirement may be to find a good enough approximation to $\boldsymbol{x}_{opt}$ i.e. to optimize f to within some tolerance δf in timescales δt. In this case, precise tracking of $\boldsymbol{x}_{opt}$ may not be necessary.)

This paper proposes a simple model of a dynamic function with moving local minima,

$$f = min\ \{f_1(\boldsymbol{x}, \boldsymbol{u}_1), f_2(\boldsymbol{x}, \boldsymbol{u}_2), \ldots, f_m(\boldsymbol{x}, \boldsymbol{u}_m)\} \quad (1)$$

where the control variables $\boldsymbol{u}_a = \{\boldsymbol{x}_a, h_a^2\}$ are defined so that f_a has a single minimum at $\boldsymbol{x}_a$, with an optimum value $h_a^2 \geq 0$ at $f_a(\boldsymbol{x}_a)$. If the functions f_a themselves have individual dynamics, f can be used to model a general dynamic environment.

A convenient choice for f_a, which allows comparison with other work on dynamic search with swarms [4, 7, 8, 9, 11], is the parabolic or sphere function in n dimensions,

$$f_a = \sum_{i=1}^{n} (x_i - x_{ai})^2 + h_a^2 \tag{2}$$

which differs from De Jong's f1 function [12] by the inclusion of a height offset h_a and a position offset x_{ia}. This model satisfies Branke's conditions for a benchmark problem (simple, easy to describe and analyze, and tunable) and is in many respects similar to his "moving peaks" benchmark problem, except that the widths of each optimum are not adjustable, and in this case we seek a *minimization* ("moving valleys") [6]. This simple function is easy to optimize with conventional methods in the static mono-modal case. However the problem becomes more acute as the number m of moving minima increases.

Our choice of f also suggests a simple interpretation. Suppose that all h_a are zero. Then f_a is the Euclidean 'squared distance' between vectors $\boldsymbol{x}$ and $\boldsymbol{x}_a$. Each local optimum position $\boldsymbol{x}_a$ can be regarded as a 'target'. Then, f is the squared distance of the nearest 'target' from the set $\{\boldsymbol{x}_a\}$ to $\boldsymbol{x}$. Suppose now that the vectors $\boldsymbol{x}$ are actually projections of vectors $\boldsymbol{y}$ in R^{n+1}, so that $\boldsymbol{y} = (\boldsymbol{x}, 0)$ and targets $\boldsymbol{y}_a$ have components ($\boldsymbol{x}_a$, h_a) in this higher dimensional space. In other words, h_a are height offsets in the $n+1$th dimension. From this perspective, f is still the squared distance to the nearest target, except that the system is restricted to R^n. For example, suppose that $\boldsymbol{x}$ is the 2-dimensional position vector of a ship, and $\{\boldsymbol{x}_a\}$ are a set of targets scattered on the sea bed at depths $\{h_a\}$. Then the square root of f at any time is the distance to the closest target and the depth of the shallowest object is $\sqrt{f(\boldsymbol{x}_{opt})}$. The task for the ship's navigator is to position the ship at $\boldsymbol{x}_{opt}$, directly over the shallowest target, given that all the targets are in independent motion along an uneven sea bed.

Since no assumptions have been made about the dynamics of the environment, the above model describes the situation where the change can occur at any time. In the periodic problem, we suppose that the control variables change simultaneously at times t_i and are held fixed at $\boldsymbol{u}_i$ for the corresponding intervals $[t_i, t_{i+1}]$:

$$\boldsymbol{u}(t) = \sum_{i} (\Theta(t_i) - \Theta(t_{i+1}))\boldsymbol{u}_i \tag{3}$$

where $\Theta(t)$ is the unit step function.

The PSO and CPSO experiments of [9] and [11] are time-dependent type I experiments with a single minimum at $\boldsymbol{x}_1$ and with $h_1 = 0$. The generalization to more difficult type I environments is achieved by introducing more local minima at positions $\boldsymbol{x}_a$, but fixing the height offsets h_a. Type II environments are easily modeled by fixing the positions of the targets, but allowing h_a to change at the end of each period. Finally, a type III environment is produced by periodically changing both $\boldsymbol{x}_a$ and h_a.

Severity is a term that has been introduced to characterize problems where the optimum position changes by a fixed amount s at a given number of iterations [4, 7]. In [7, 11] the optimum position changes by small increments along a line. However, Blackwell and Bentley have considered more severe dynamic systems whereby the optimum position can jump randomly within a target cube T which is of dimension equal to twice the dynamic range v_{max} [9]. Here severity is extended to include dynamic systems where the target jumps may be for periods of very short duration.

4 PSO and CPSO Algorithms

Table 1 shows the particle update algorithm. The PSO parameters g_1, g_2 and w govern convergence. The electrostatic acceleration $\boldsymbol{a}_i$, parameterized by p_{core}, p and Q_i, is

$$\boldsymbol{a}_i = \sum_{j \neq i} \frac{Q_i Q_j}{\left|r_{ij}\right|^3} \boldsymbol{r}_{ij}, \quad p_{core} < r_{ij} < p, \quad \boldsymbol{r}_{ij} = \boldsymbol{x}_i - \boldsymbol{x}_j \tag{4}$$

The PSO and CPSO search algorithm is summarized below in Table 2. To begin, a swarm of M particles, where each particle has n-dimensional position and velocity vectors $\{\boldsymbol{x}_i, \boldsymbol{v}_i,\}$, is randomized in the box $T = D^n = [-v_{max}, v_{max}]^n$ where D is the 'dynamic range' and v_{max} is the clamping velocity. A set of period durations $\{t_i\}$ is chosen; these are either fixed to a common duration, or chosen from a uniform random distribution. A single iteration is a single pass through the loop in Table 2.

Denoting the best value position and value found by the swarm as $\boldsymbol{x}_{gb}$ and f_{gb}, change detection is simply invoked by comparing $f(\boldsymbol{x}_{gb})$ with f_{gb}. If these are not equal, the inference is that f has changed since f_{gb} was last evaluated. The response is to re-randomize a fraction of the swarm in T, and to re-set f_{gb} to $f(\boldsymbol{x}_{gb})$. The detection and response algorithm is only applied to neutral swarms.

The best position attained by a particle, $\boldsymbol{x}_{pb,i}$, is updated by comparing $f(\boldsymbol{x}_i)$ with $f(\boldsymbol{x}_{pb,i})$: if $f(\boldsymbol{x}_i) < f(\boldsymbol{x}_{pb,i})$, then $\boldsymbol{x}_{pb,i} \leftarrow \boldsymbol{x}_i$. Any new $\boldsymbol{x}_{pb,i}$ is then tested against $\boldsymbol{x}_{gb}$, and a replacement is made, so that at each particle update $f(\boldsymbol{x}_{gb}) = min\{f(\boldsymbol{x}_{pb,i})\}$. This specifies *update best*(i).

Table 1. The particle update algorithm

update particle(i)
$\boldsymbol{v}_i \leftarrow w\boldsymbol{v}_i + g_1(\boldsymbol{x}_{pb,i} - \boldsymbol{x}_i) + g_2(\boldsymbol{x}_{gb} - \boldsymbol{x}_i) + \boldsymbol{a}_i$
if $\lvert\boldsymbol{v}_i\rvert > v_{max}$
$\quad\quad \boldsymbol{v}_i \leftarrow (v_{max} / \lvert\boldsymbol{v}_i\rvert)\, \boldsymbol{v}_i$
$\boldsymbol{x}_i \leftarrow \boldsymbol{x}_i + \boldsymbol{v}_i$

Table 2. Search algorithm for charged and neutral particle swarm optimization

```
(C)PSO search
initialize swarm { x_i, v_i} and periods{t_j}
loop:
    if t = t_j
        update function
    if (neutral swarm)
        detect and respond to change
    for i = 1 to M
        update best (i)
        update particle(i)
    endfor
    t←t+1
until stopping criterion is met
```

5 Experiment Design

Twelve experiments of varying severity were conceived, for convenience arranged in three groups. The parameters and specifications for these experiments are summarized in Tables 3 and 4. In each experiment, the dynamic function has two local minima at $\boldsymbol{x}_a$, $a = 1, 2$; the global minimum is at $\boldsymbol{x}_2$. The value of f at $\boldsymbol{x}_1$ is fixed at 100 in all experiments. The duration of the function update periods, denoted Δ, is either fixed at 100 iterations, or is a random integer between 1 and 100. (For simplicity, random variables drawn from uniform distribution with limits a, b will be denoted x ~ [a, b] (continuous distribution) and x ~ [a…b] (discrete distribution).

In the first group (A) of experiments, numbers 1 – 4, $\boldsymbol{x}_2$ is moved randomly in T ('spatially severe') or is moved randomly in a smaller box 0.1T. The optimum value, $f(\boldsymbol{x}_2)$, is fixed at 0. These are all type I experiments, since the optimum location moves, but the optimum value is fixed. Experiments 3 and 4 repeat the conditions of 1 and 2 except that $\boldsymbol{x}_2$ moves at random intervals ~ [1…100] (temporally severe).

Experiments 5 – 8 (Group B) are type II environments. In this case, $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ are fixed at $\pm\boldsymbol{r}$, along the body diagonal of T, where $\boldsymbol{r} = (\boldsymbol{v}_{max}/3)\ (1, 1, 1)$. However, $f(\boldsymbol{x}_2)$ varies, with h_2 ~ [0, 1], or h_2 ~ [0, 100]. Experiments 7 and 8 repeat the conditions of 5 and 6 but for high temporal severity.

In the last group (C) of experiments (9 – 12), both $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ jump randomly in T. In the type III case, experiments 11 and 12, $f(\boldsymbol{x}_2)$ varies. For comparison, experiments 9 and 10 duplicate the conditions of 11 and 12, but with fixed $f(\boldsymbol{x}_2)$. Experiments 10 and 12 are temporally severe versions of 9 and 11.

Each experiment, of 500 periods, was performed with neutral, atomic (i.e. half the swarm is charged) and fully charged swarms (all particles are charged) of 20 particles (M = 20). In addition, the experiments were repeated with a random search algorithm, which simply, at each iteration, randomizes the particles within T. A spatial dimension of $n = 3$ was chosen. In each run, whenever random numbers are required for target positions, height offsets and period durations, the same sequence of pseudo-random numbers is used, produced by separately seeded generators. The initial swarm configuration is random in T, and the same configuration is used for each run.

Table 3. Spatial, electrostatic and PSO Parameters

Spatial				**Electrostatic**			**PSO**	
v_{max}	n	M	T	p_{core}	p	Q_i	g_1, g_2	w
32	3	20	$[-32,32]^3$	1	$2\sqrt{3}v_{max}$	16	~[0,1.49]	~[0.5, 1]

Table 4. Experiment Specifications

Group	**Expt**	**Targets** $\{\boldsymbol{x}_1, \boldsymbol{x}_1\}$	**Local Opt** $\{f(\boldsymbol{x}_1), f(\boldsymbol{x}_2)\}$	**Period** Δ
A	1	$\{\boldsymbol{O}, \sim 0.1T\}$	{100, 0}	100
	2	$\{\boldsymbol{O}, \sim T\}$		
	3	$\{\boldsymbol{O}, \sim 0.1T\}$		~[1, 100]
	4	$\{\boldsymbol{O}, \sim T\}$		
B	5	$\{\boldsymbol{O}-\boldsymbol{r}, \boldsymbol{O}+\boldsymbol{r}\}$	{100, ~[0, 1]}	**100**
	6		{100,~[0,100]}	
	7		{100, ~[0, 1]}	**~[1, 100]**
	8		{100,~[0,100]}	
C	9	$\{\sim T, \sim T\}$	{100, 0]}	**100**
	10			**~[1,100]**
	11		{100,~[0,100]}	**100**
	12			**~[1,100]**

The search (C)PSO algorithm has a number of parameters (Table 3) which have been chosen to correspond to the values used in previous experiments [5, 9, 11]. These choices agree with Clerc's analysis for convergence [13]. The spatial and electrostatic parameters are once more chosen for comparison with previous work on charged particle swarms [9]. An analysis that explains the choice of the electrostatic parameters is given in [14]. Since we are concerned with very severe environments, the response strategy chosen here is to randomize the positions of 50% of the swarm [11]. This also allows for comparisons with the atomic swarm which maintains a diverse population of 50% of the swarm.

6 Results and Analysis

The chief statistic is the ensemble average best value, $<f(\boldsymbol{x}_2) - f_{gb}>$; this is positive and bounded by zero. A further statistic, the number of 'successes', $n_{successes,}$, was also collected to aid analysis. Here, the search is deemed a success if $\boldsymbol{x}_{gb}$ is closer, at the end of each period, to target 2 (which always has the lower value of f) than it is to target 1. The results for the three swarms and for random search are shown in Figs 1 and 2. The light grey boxes in Figure 1, experiment 6, indicate an upper bound to the ensemble average due to the precision of the floating-point representation: for these runs, $f(\boldsymbol{x}_2) - f_{gb} = 0$ at the end of each period, but this is an artifact of the finite-precision arithmetic.

Group A. Figure 1 shows that all swarms perform better than random search except for the neutral swarm in spatially severe environments (2 and 4) and the atomic swarm in a spatially and temporally severe environment (4). In the least severe environment (1), the neutral swarm performs very well, confirming previous results. This swarm has the least diversity and the best exploitation. The order of performance for this experiment reflects the amount of diversity; neutral (least diversity, best), atomic, fully charged, and random (most diversity, worst). When environment 1 is made temporally severe (3), all swarms have similar performance and are better than random search. The implication here is that on average the environment changes too quickly for the better exploitation properties of the neutral swarm to become noticeable. Experiments 2 and 4 repeat the conditions of 1 and 2, except for higher spatial severity. Here the order of performance amongst the swarms is in increasing order of diversity (fully charged best and neutral worst). The reason for the poor performance of the neutral swarm in environments 2 and 4 can be inferred from the success data. The success rate of just 5% and ensemble average close to 100 ($= f(\boldsymbol{x}_1)$) suggests that the neutral swarm often gets stuck in the false minimum at $\boldsymbol{x}_1$. Since f_{gb} does not change at $\boldsymbol{x}_1$, the adapted swarm cannot register change, does not randomize, and so is unlikely to move away from $\boldsymbol{x}_1$ until $\boldsymbol{x}_2$ jumps to a nearby location. In fact the neutral swarm is worse than random search by an order of magnitude. Only the fully charged swarm out-performs random search appreciably for the spatially severe type I environments (2 and 4) and this margin diminishes when the environment is temporally severe too.

Group B. Throughout this group, all swarms are better than random and the number of successes shows that there no problems with the false minimum. The swarm with the least diversity and best exploitation (neutral) does best since the optimum location

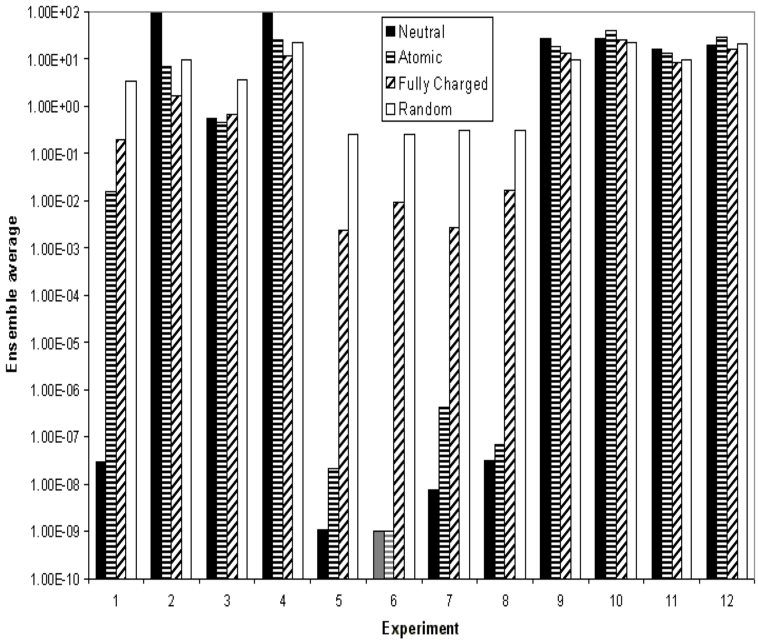

Fig. 1. Ensemble average $<f(\mathbf{x}_?)-f_{ob}>$ for all experiments

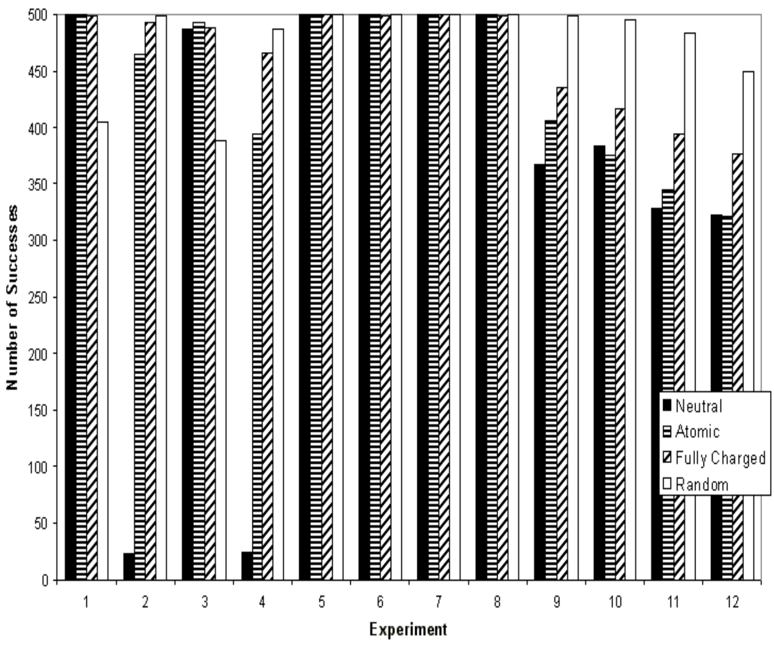

Fig. 2. Number of successes $n_{successes}$ for all experiments

does not change from period to period. The effect of increasing temporal severity can be seen by comparing 7 to 5 and 8 to 6. Fully charged and random are almost unaffected by temporal severity in these type II environments, but the performance of the neutral and atomic swarms worsens. Once more the explanation for this is that these are the only two algorithms which can significantly improve their best position over time because only these two contain neutral particles which can converge unimpeded on the minimum. This advantage is lessened when the average time between jumps is decreased. The near equality of ensemble averages for random search in 5 and 6, and again in 7 and 8, is due to the fact that random search is not trying to improve on a previous value – it just depends on the closest randomly generated points to $\boldsymbol{x}_2$ during any period. Since $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ are fixed, this can only depend on the period size and not on $f(\boldsymbol{x}_2)$.

Group C. The ensemble averages for the four experiments in this group (9-12) are broadly similar but the algorithm with the most successes in each experiment is random search. However random search is not able to exploit any good solution, so although the swarms have more failures, they are able to improve on their successes producing ensemble averages close to random search. In experiments 9 and 10, which are type I cases, all swarms perform less well than random search. These two experiments differ from environments 2 and 4, which are also spatially severe, by allowing the false minimum at $\boldsymbol{x}_1$ to jump as well. The result is that the performance of the neutral swarm improves since it is no longer caught by the false minimum at $\boldsymbol{x}_1$; the number of successes improves from less than 25 in 2 and 4, to over 350 in 9 and 10. In experiments 11 and 12 (type III) when f_{opt} changes in each period, the fully charged swarm marginally out-performs random search. It is worth noting that 12 is a very extreme environment: either minimum can jump by arbitrary amounts, on any time scale, and with the minimum value varying over a wide range. One explanation for the poor performance of all swarms in 9 and 10 is that there is a higher penalty ($\langle f(\boldsymbol{x}_1) - f_{opt} \rangle = 100$) for getting stuck on the false minimum at $\boldsymbol{x}_1$, than the corresponding penalty in 11 and 12 ($\langle f(\boldsymbol{x}_1) - f_{opt} \rangle = 50$). The lower success rate for all swarms compared to random search supports this explanation.

7 Conclusions

A dynamic environment can present numerous challenges for optimization. This paper has presented a simple mathematical model which can represent dynamic environments of various types and severity. The neutral particle swarm is a promising algorithm for these problems since it performs well in the static case, and can be adapted to respond to change. However, one draw back is the arbitrary nature of the detection and response algorithms. Particle swarms with charge need no further adaptation to cope with the dynamic scenario due to the extended swarm shape. The neutral and two charged particle swarms have been tested, and compared with random search, with twelve environments which are classified by type. Some of these environments are extreme, both in the spatial as well as the temporal domain.

The results support the intuitive idea that type II environments (those in which the optimum location is fixed, but the optimum value may vary) present few problems to evolutionary methods since a population diversity is not important. In fact the algorithm with the lowest diversity performed best. Increasing temporal severity diminishes the performance of the two swarms with neutral particles, but does not affect the fully charged swarm.

However, environments where the optimum location can change (types I and III) are much harder to deal with, especially when the optimum jumps can be to an arbitrary point within the search space, and can happen at very short notice. This is the dynamic equivalent of the needle in a haystack problem. A type I environment has been identified which poses considerable problems for the adapted PSO algorithm: a stationary false minimum and a mobile true minimum with large spatial severity. There is a tendency for the neutral swarm to become trapped by the false minimum. In this case, the fully charged swarm is the better option.

Finally, the group C environments proved to be very challenging for all swarms. These environments are distinguished by two spatially severe minima with a large difference in function value at these minima. In other words, there is a large penalty for finding the false minimum rather than the true minimum. All swarms struggled to improve upon random search because of this trap.

Despite this, all swarms have been shown, for dynamic parabolic functions, to offer results comparable to random search in the worst cases, and considerably better than random in the more benign situations. As with static search problems, if some prior knowledge of the dynamics is known, a preferable algorithm can be chosen. According to the classification of Eberhart and Wu [7, 11], and for the examples studied here, the adapted neutral swarm is the best performer for mild type I and II environments. However, it can be easily fooled in type I and III environments where a false minimum is also dynamic. In this case, the charged swarms are better choices. As the environment becomes more extreme, charge, which is a diversity increasing parameter, becomes more useful. In short, if nothing is known about an environment, the fully charged swarm has the best average performance.

It is possible that different adaptations to the neutral swarm can lead to better performance in certain environments, but it remains to be seen if there is a single adaptation which works well over a range of environments. On the other hand, the charged swarm needs no further modification since the collision avoiding accelerations ensure exploration the space around a solution.

References

1. Kennedy J. and Eberhart, R.C.: Particle Swarm Optimization. Proc of the IEEE International Conference on Neural Networks IV (1995) 1942–1948
2. Eberhart R.C. and Shi Y.: Particle swarm optimization: Developments, applications and resources. Proc Congress on Evolutionary Computation (2001) 81–86
3. Saunders P.T.: An Introduction to Catastrophe Theory. Cambridge University Press (1980)

4. Angeline P.J.: Tracking extrema in dynamic environments. Proc Evolutionary Programming IV. (1998) 335–345
5. Bäck T.: On the behaviour of evolutionary algorithms in dynamic environments. Proc Int. Conf. on Evolutionary Computation. (1998) 446–451
6. Branke J.: Evolutionary algorithms for changing optimization problems. Proc Congress on Evolutionary Computation. (1999) 1875–1882
7. Eberhart R.C. and Shi Y.: Tracking and optimizing dynamic systems with particle swarms. Proc Congress on Evolutionary Computation. (2001) 94–97
8. Carlisle A. and Dozier G.: Adapting particle swarm optimization to dynamic environments. Proc of Int Conference on Artificial Intelligence. (2000) 429–434
9. Blackwell and Bentley P.J.: Dynamic search with charged swarms. Proc Genetic and Evolutionary Computation Conference. (2002) 19–26
10. Blackwell and Bentley P.J.: Don't push me! Collision avoiding swarms. Proc Congress on Evolutionary Computation. (2002) 1691–1696
11. Hu X. and Eberhart R.C.: Adaptive particle swarm optimization: detection and response to dynamic systems. Proc Congress on Evolutionary Computation. (2002) 1666–1670
12. De Jong K: An analysis of the behavior of a class of genetic adaptive systems. PhD thesis, University of Michigan (1975)
13. Clerc M.: The swarm and the queen: towards a deterministic and adaptive particle swarm optimization. Proc Congress on Evolutionary Computation. (1999) 1951–1957
14. Blackwell and Bentley P.J.: Improvised Music with Swarms, Proc Congress on Evolutionary Computation. (2002) 1462–1467