Dynamic Search With Charged Swarms.

Conference Paper · January 2002

Source: DBLP

CITATIONS
225

READS
523

2 authors, including:

Peter J. Bentley
University College London
331 PUBLICATIONS **7,743** CITATIONS

SEE PROFILE

Dynamic Search with Charged Swarms

T.M.Blackwell

Department of Computer Science
University College London
Gower Street,
London, UK
tim@theHotandtheCool.co.uk

P. J. Bentley

Department of Computer Science
University College London
Gower Street,
London, UK
P.Bentley@cs.ucl.ac.uk

Abstract

Two novel particle swarm optimization (PSO) algorithms are used to track and optimize a 3-dimensional parabolic benchmark function where the optimum location changes randomly and with high severity. The new algorithms are based on an analogy of electrostatic energy with charged particles. For comparison, the same experiment is performed with a conventional PSO algorithm. It is found that the best strategy for this particular problem involves a combination of neutral and charged particles.

1 INTRODUCTION

Particle Swarm Optimization (PSO) is a population based evolutionary technique applied to optimization problems. It differs from other population approaches such as genetic algorithms, by the inclusion of a solution (or particle) velocity, which moves the position of the solution in the space of all possible solutions, rather than relying on recombination of existing solutions. Linear spring forces govern the dynamics of the population (or swarm); each particle is attracted to its previous best position, and to the global best position attained by the swarm, where fitness is quantified by the value of a 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 (Eberhart and Shi 2001a).

The real world is rarely static, however, and many systems will require frequent re-optimization due to system and/or environmental change. The problem of re-scheduling system resources is an important example of this. One important and implicit constraint is the requirement to balance the desired error of the solution with the need to be prepared to respond rapidly to change. For example, to achieve a low error will require a large number of iterations/generations, and will leave the evolutionary population well adapted to that situation. But system and environment change may occur on short time-scales and may be large enough to leave the population ill-adapted to the new problem, so that a solution considered good enough may be hard to find within this time-scale.

This work addresses these issues with the use of two novel swarm algorithms. These algorithms are tested and compared with the conventional PSO algorithm for an extreme search problem wherein the optimum location (solution) is randomized within a box representing the entire dynamic range.

2 BACKGROUND

Eberhart and Shi (2001b) have applied the conventional PSO algorithm to some dynamic search and optimization problems. In their experiments, they use a time-scale of 100 iterations, and choose as a benchmark the (3-dimensional) parabolic function and the sphere function in 10-dimensions. The optimum location of these functions was moved along a line by increments of 0.2% and 1% of the dynamic range, with each change occurring at 100 iterations. It was found that, under these conditions, the PSO algorithm performed at least as well as other evolutionary techniques (Angeline 1997, Bäck 1998).

One drawback noted by Eberhart and Shi is the lack of a strategy for dealing with a wide variety of change. One possibility is to randomize the swarm when a change is detected. In their work (2001b), the particle positions are retained, but the personal and global best positions are calculated with respect to the new optimum location. Another possibility would be to randomize the swarm when a change is detected. In general, a good strategy is needed that can account for chaotic rather than linear change, and for change that is commensurate with the entire range of the dynamic variables, and not just limited to one per cent of this range.

This work investigates the capabilities of two novel swarm algorithms to overcome an extreme problem of this type. The two new algorithms were originated by the authors in quite a different context: the problem of artificial improvised music (Blackwell and Bentley

2002a). It was demonstrated that particle swarms can, if suitably interpreted as music, generate interesting melodies. Moreover, they can also interact with an external musician. External audio events are interpreted and placed in the search space, and become targets or attractors for the swarm. These targets may change on very small time-scales, and by large amounts. It was found in this work that inter-particle repulsion or "collision-avoidance" balances the target attractions and leads to an extended swarm that follows this change well. Various features of the algorithm have been reported in a subsequent paper and the suggestion made that they may have relevance to optimization problems (Blackwell and Bentley, 2002b).

The particular form of the repulsive force we have introduced is identical to the familiar electrostatic inverse square law between identically charged particles. In this paper we consider two different swarms: the first is composed entirely of identically charged particles, and the second has an equal number of charged and 'neutral' particles. Neutral particles do not experience the repulsive force. (Within this electrostatic analogy, it could be said that conventional PSO algorithms concern only neutral particles.) The idea is that the neutral particles will gather around the global best position (as if in a nucleus) whilst the charged particles will continue to explore the solution space as they orbit the nucleus. Hence there will be a balance between exploration and exploitation. This type of swarm could be termed 'atomic' since it has much in common with models of the atom. As such, it moves away from the original idea of an insect swarm or avian flock which inspired much of the early work on particle swarms.

3 THE PROBLEM

The dynamic problem investigated in this work is to find the global minimum f(**0**) of some function f($\mathbf{x}$-$\mathbf{x}_{opt}$) where $\mathbf{x}_{opt}$ is the optimum location. For dynamic search, $\mathbf{x}_{opt}$ = $\mathbf{x}_{opt}$(t), where t is an iteration counter although it could be a time variable as determined by the actual dynamic environment. Eberhart and Shi (2001b) hold $\mathbf{x}_{opt}$ fixed for 100 iterations at a time, and $\mathbf{x}_{opt}$ varies in increments of s**1,** where **1** is the unit vector in n-dimensions (linear change) for s = 0.1 and s = 0.5. The dynamic range of the variables is [-50, 50] in each dimension.

4 PARTICLE DYNAMICS

Within the PSO methodology, the particle dynamics are determined by an update rule which modifies particle velocities. New positions are then found by adding the updated velocity to the current position. The particle update algorithm used in this work is given by the application of three simple steps:

$$\mathbf{v}_i \leftarrow w\mathbf{v}_i + c_1r_1(\mathbf{x}_{pb,i} - \mathbf{x}_i) + c_2r_2(\mathbf{x}_{gb}-\mathbf{x}_i) \quad (1)$$

$$\textbf{if } (\,|\mathbf{v}_i| > v_{max})\ \ \mathbf{v}_i \leftarrow (v_{max} / |\mathbf{v}_i|\,)\ \mathbf{v}_i \quad (2)$$

$$\mathbf{x}_i \leftarrow \mathbf{x}_i + \mathbf{v}_i \quad (3)$$

In these rules, i is a particle label and each particle has a position $\mathbf{x}$ and a velocity $\mathbf{v}$ (n-dimensional vectors). The inertia weight w, and spring constants c_1 and c_2 are the adjustable parameters of the algorithm. r_1 and r_2 are random numbers drawn from the unit interval, $r_1, r_2 \in [0,1]$. $\mathbf{x}_{pb,i}$ is the best position attained by particle i and the global best location $\mathbf{x}_{gb}$ is the best position attained by any particle.

Rule (1) adds the particle accelerations from the spring forces to a damped velocity w$\mathbf{v}_i$. Rule (2) clamps the velocity to the dynamic range $[-v_{max}, v_{max}]$, which serves to limit the position increment applied in Rule (3). Notice that our rule (2) implements spherically symmetric velocity clamping, whereas other PSO algorithms clamp the velocity to a box. Since the following experiments involve qualitative observations on the spatial distribution of particles at any iteration, it is necessary to preserve spherical symmetry in the update rules.

Table 1: Search algorithm

Initialize a swarm $\{\mathbf{x}_i, \mathbf{v}_i\}$, i =1,…M, with $\mathbf{x}_i \in [0, x_{max}]^n$ and $\mathbf{v}_i \in [-v_{max}, v_{max}]^n$

Set all personal best positions to $\mathbf{x}_{pb,i}$ to $\mathbf{x}_i$

t←0

do:

  for i = 1 to M

    if f($\mathbf{x}_i$ -$\mathbf{x}_{opt}$) < f($\mathbf{x}_{pb,i}$ -$\mathbf{x}_{opt}$)

      then $\mathbf{x}_{pb,i}$← $\mathbf{x}_i$

    if f($\mathbf{x}_{pb,i}$ -$\mathbf{x}_{opt}$) < f($\mathbf{x}_{gb}$-$\mathbf{x}_{opt}$)

      then $\mathbf{x}_{gb}$← $\mathbf{x}_{pb,i}$

  endfor

  if (t%100 = 0)

    then $\mathbf{x}_{opt} \in [(x_{max}/2)\text{-}L/2, (x_{max}/2)+L/2]^3$

  for i = 1 to M

    Apply particle update algorithm (1) – (3)

  endfor

  t←t+1

until stopping criterion is met

This clamping is a constraint on global exploration. The balance between this and local exploitation of good solutions is given by the inertial weight. In the non-dynamic case it is advantageous to reduce w from 1 down to near zero during the course of training run, since this allows full exploitation of possible good solutions (Eberhart and Shi 2001a). However, in the dynamic case it cannot be predicted whether exploration or exploitation is needed at any given time. With these factors in mind, Eberhart and Shi (2001b) used a value of 1.494 for the spring constants and a random inertia weight w ∈ [0.5, 1]. These values were chosen to agree, on the average,

with Clerc's analysis for convergence (Clerc 1999), and to provide a balance between exploration and exploitation.

The PSO algorithm for the dynamic problem investigated here is given in table 1.

In order to introduce the notion of charge – and hence collision avoidance – all that needs to be done is to modify the rule for particle accelerations, rule (1). The grounds for this extension, and a full description of the effects of the various parameters on particle motion, are given in (Blackwell, 2001) and (Blackwell and Bentley, 2002b). In those studies, an additional acceleration towards the swarm centre was also implemented, but this is not used here.

The necessary amendment to the particle update algorithm is an extra particle acceleration $\mathbf{a}_i$ given by

$$\mathbf{a}_i = \sum_{j \neq i} \frac{Q_i Q_j}{r_{ij}^3} \mathbf{r}_{ij}, \qquad p_{core} < r_{ij} < p \qquad (4)$$

where $\mathbf{r}_{ij} = \mathbf{x}_i - \mathbf{x}_j$, $r_{ij} = |\mathbf{x}_i - \mathbf{x}_j|$ and each particle has a charge of magnitude Q_i. Neutral particles are assigned a charge $Q_i = 0$ and so will not contribute to the sum in (4). A charged particle i will have $Q_i > 0$ and will experience the repulsive effects from all other charged particles $j \neq i$. Repulsion is only experienced for separations within the shell $p_{core} < r_{ij} < p$. The lower cut-off p_{core} is a safeguard against the singularity of the inverse square law. The upper cut off p is a tunable parameter allowing the domain of influence of the repulsion to be controlled.

The charged particle update algorithm, which replaces the particle update algorithm within the search algorithm (Table 1), is given by replacing rule (1) with:

$$\mathbf{v}_i \leftarrow w\mathbf{v}_i + c_1 r_1(\mathbf{x}_{pb,i} - \mathbf{x}_i) + c_2 r_2(\mathbf{x}_{gb} - \mathbf{x}_i) + \mathbf{a}_i \quad (1')$$

It is worth noting that since the particles are updated in turn (i.e. from i = 1 to i = M), contributions to $\mathbf{a}_i$ can involve non-updated particle positions (j > i) as well as updated positions (j < i). This was found to give better avoidance in earlier experiments.

5 EXPERIMENTS

The experiments were conducted on the parabolic function in n = 3 dimensions, $f(\mathbf{x}-\mathbf{x}_{opt}) = (\mathbf{x}-\mathbf{x}_{opt}) \bullet (\mathbf{x}-\mathbf{x}_{opt})$.

The task is made dynamic by placing $\mathbf{x}_{opt}$ in a cube of side L and then randomly re-positioning it to another point in this cube every 100 iterations. The severity *s* therefore varies randomly from zero up to $(\sqrt{3})$L. With L set to $2v_{max}$, this gives a severity of up to $2\sqrt{3}$ times the dynamic range. The parameter x_{max} solely determines the initial distribution of the particle velocities and positions and plays no part in subsequent updates after the first jump in $\mathbf{x}_{opt}$. The values of the spatial parameters set out in Table 2.

Table 2: Spatial parameters

L	x_{max}	v_{max}	M
64	128	32	20

The values of the electrostatic parameters p_{core}, p and Q are set out in Table 3.

Table 3: Electrostatic parameters

p_{core}	p	Q
1	$\sqrt{3}x_{max}$	16

The values of the PSO parameters w, c_1 and c_2 are those used by Eberhart and Shi (2001b). The inertia weight w varies randomly between 0.5 and 1.0, so that the mean 0.75 is close to the Clerc constriction factor 0.729 (Clerc 1999). The spring constants c_1 and c_2 are set to 1.494, also in accordance with Clerc's analysis.

In all these experiments, 50 optimum jumps are made, (or 5000 iterations of the system). In addition to the numerical data produced by these experiments, a three dimensional animation was set up which enabled a qualitative assessment of the three swarms.

For comparison, four experiments were performed:

I Neutral swarm.

The first experiment uses the conventional PSO algorithm, which is implemented by setting the charge on all 20 particles to zero (i.e. each particle is neutral).

II Charged swarm.

In the second experiment, all 20 particles carry the same charge, Q. In other words, all particles experience repulsive forces from the other particles.

III Atomic swarm.

The third experiment evaluates the *atomic swarm*, where 10 particles have charge Q and the remaining half are neutral.

IV Neutral swarm, one optimum jump.

The fourth experiment is identical to Experiment 1 except that just 200 iterations were allowed. The positions and velocities of the particles were saved to file for analysis of individual particle motion.

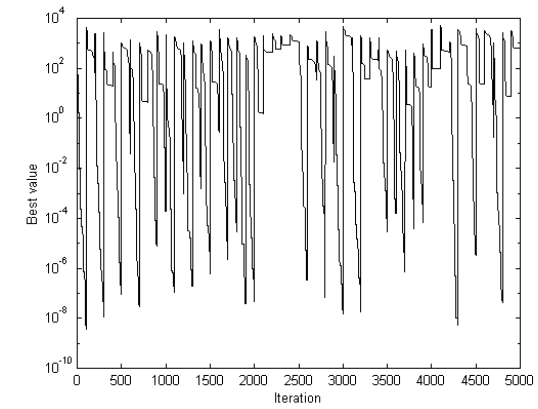

Figure 1: Neutral swarm

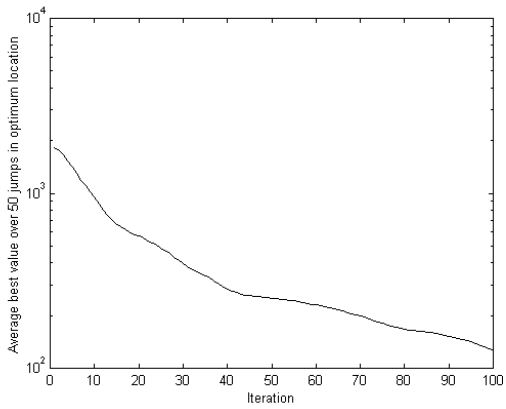

Figure 2: Neutral swarm, average best values per 100 iterations.

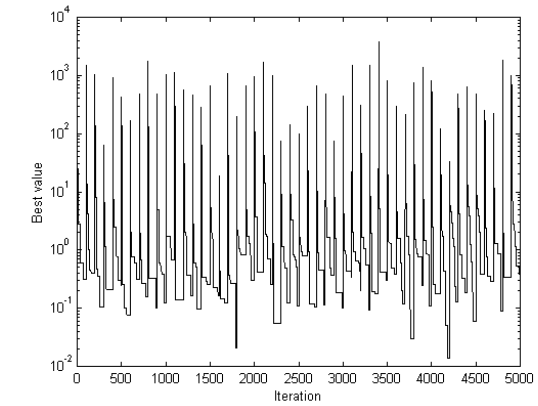

Figure 3: Charged swarm

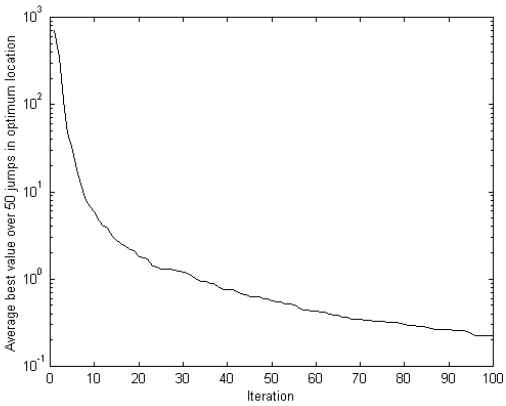

Figure 4: Charged swarm, average best values per 100 iterations.

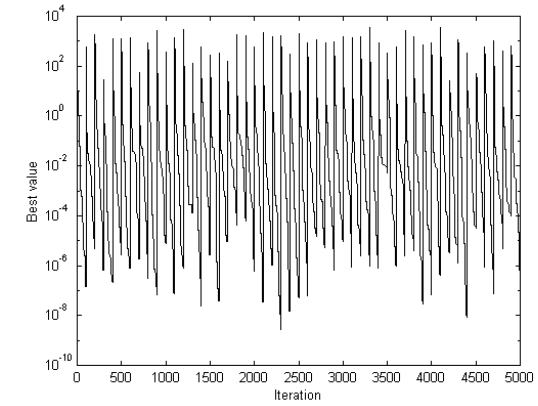

Figure 5: Atomic swarm

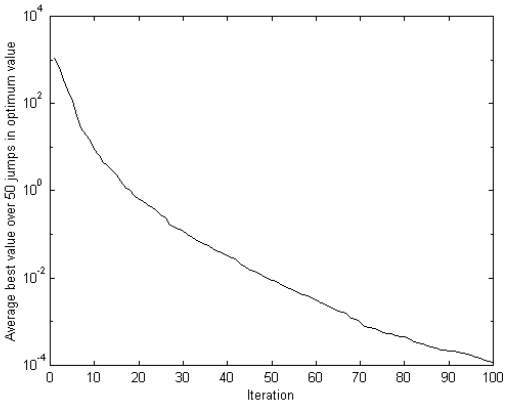

Figure 6:Atomic swarm, average best values per 100 iterations.

6 RESULTS

The results for the experiments on the neutral, charged and atomic swarms are shown in Figures 1 to 6. In each case, two graphs have been prepared, plotted as a function of iteration number t: the best value found by the swarm, $f(\mathbf{x}_{gb} - \mathbf{x}_{opt})$, and the average best value over the 50 problem optimum jumps (i.e., average best per 100 iterations, from optimum jump to the iteration before the next optimum jump).

6.1 EXPERIMENT I: NEUTRAL SWARM

The best value attained after 100 iterations, i.e. just before the first optimum jump, is of the order of 10^{-8}. This is at least two orders of magnitude lower than the best value obtained in the first 100 iterations of an equivalent experiment (Eberhart and Shi 2001b figure 1, p98), although comparable to the best values obtained in subsequent optimum positions. The discrepancy is presumably due to initial conditions, although the spherically symmetric clamping rule may play a part.

However, the results depart significantly at the first optimum jump, due to the increased severity. The remarkable feature of Figure 1 is the small spikes at the optimum jump followed by a leveling of the graph for some tens of iterations, after a short fall. This plateau in best values sometimes then drops before the next optimum jump, but often does not. For example, there is a run from t = 2100 to 2500 where the initial short fall is not improved upon. The plot of the averages over the 50 optimum jumps shows an average best of 125 at 100 iterations. The slope at this point is -3.2, indicating only a slow improvement (3% per iteration) with increasing iteration.

The 3D animations showed a very unusual feature. For the first 100 iterations, the particles were clumped very closely around the optimum. At the optimum jump, however, the particles moved along a line in the general direction of the new optimum, and then began to oscillate along this line about a point close, but not adjacent to the new optimum. After some tens of iterations the oscillations would cease and the particles would begin to swarm towards the new optimum, although they might not reach it before the next jump. This behavior was repeated invariably at each optimum jump, and in repeats of this experiment.

6.2 EXPERIMENT II: CHARGED SWARM

Figure 3 shows the best values over the 5000 iterations. By comparison with Figure 1, the spikes are now long, showing an improved best value by a factor of 10^3 after just a few iterations at each jump. The leveling out now occurs at a much smaller best value, a feature illustrated in Figure 4 which shows the average best values. In Figure 4, the lowest average best value obtained is 0.226, and the slope at this point is -2.10×10^{-3} showing an improvement of 1% per iteration at this point.

The animations revealed typical swarming behavior: at each optimum jump the swarm moved towards the new optimum, with irregular motion about the swarm centre. After a few iterations the swarm centre was coincident with the optimum and the particle motion continued to be chaotic and spherically symmetric about this point, with particles amplitudes of some tens of units. These pictures agreed with previous swarm experiments (Blackwell and Bentley, 2001b).

6.3 EXPERIMENT III: ATOMIC SWARM.

Once more, the plot of best values, Figure 5, shows spikes at each optimum jump, but the spikes drop to a much lower best value, in the range 10^{-4} to 10^{-8} in 49 of the 50 jumps. The figure does not show the plateaus that are a feature of Figures 1 and 3. The plot of average best values, Figure 6, shows a much improved average best at 100 iterations of 1.12×10^{-4}, with a slope of -1.21×10^{-5} or 11% of the best value per iteration at this point. The average global best just before the next optimum jump is at least 6 orders of magnitude better than the neutral swarm and about 2000 times better than the charged swarm.

In order to distinguish charged from neutral particles for the purposes of the animation, the particles were colored red (charged) and blue (neutral). At each optimum jump, the animations displayed the particles moving in an irregular swarming motion towards the optimum, followed by a long period where the blue neutral particles clumped around the optimum, moving ever slower with very small amplitude, surrounded with a 'cloud' of charged red particles, moving much like the charged swarm described in II. The picture was very reminiscent of representations of an atomic nucleus surrounded by an electron cloud.

6.4 EXPERIMENT IV: NEUTRAL SWARM, ONE OPTIMUM JUMP.

A further experiment was conducted on the neutral swarm to give greater insight on the linear non-swarming behavior of the particles just after the optimum jump, as observed in the animation. Just two hundred iterations were completed, allowing a single optimum jump to occur.

At t =100, the 20 particles were clumped very tightly around x_o, and moving slowly. Between t = 100 and t = 120, all the particles followed a very similar trajectory. For this reason, some results for just a single particle, particle 0, will be presented. Table 4 shows x, y and z components of the optimum location, global best location and position and velocity of particle 0 at iteration 99, just before the effects of the optimum jump have influenced the particle dynamics.

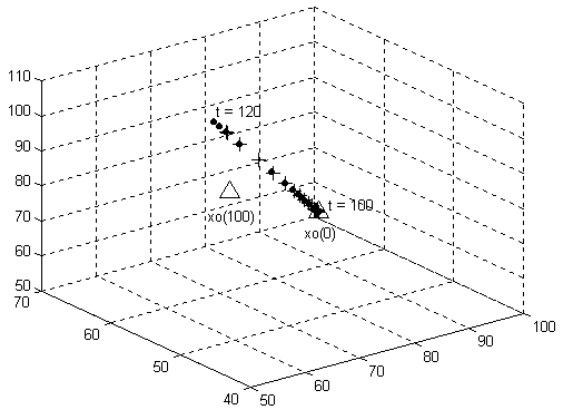

Figure 7: Position of particle 0 for t = 100 to t = 120

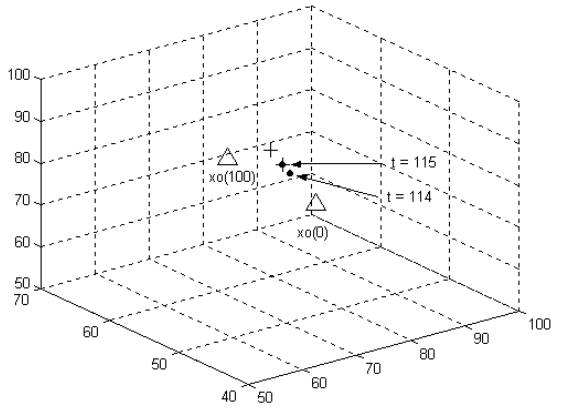

Figure 8: Snapshots at t = 114 and t = 115

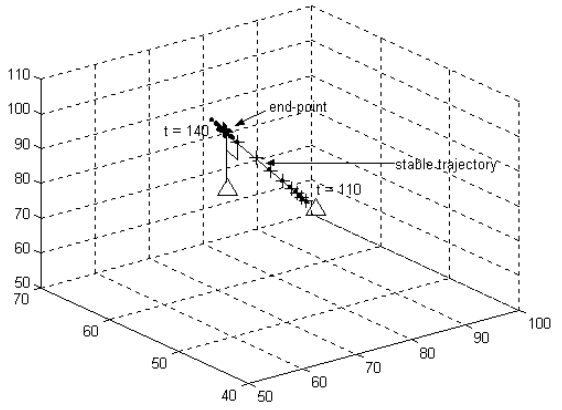

Figure 9: Position of particle 0 for t = 110 to t = 140

Table 4: Opt., best & particle 0 components at t = 99.

	$x_{opt}(100)$	$x_{gb}(100)$	$x_0(100)$	$v_0(100)$
x	95.71089	95.710884	95.5921	0.61443764
y	65.98201	65.98204	65.87358	0.52882737
z	56.992935	56.992916	57.107887	-0.58677804

Figure 7 shows the positions of particle 0 (circles) and the global best (+'s) for iterations t = 100 to t = 120. The optimum location is depicted with a triangle. Figure 8 shows just two snapshots at t = 114 and t = 115. Finally, Figure 9 shows the position of particle 0 and global best between iterations 110 and 140. For the purpose of the subsequent analysis, a line showing the stable trajectory and its end point has been marked on the figure.

7 ANALYSIS

The strange behavior of the neutral swarm just after the optimum jump is the crucial difference between Experiments I and II. This analysis section will start with a possible explanation for this phenomenon.

The situation for the neutral swarm just at and after the optimum jump must be studied, and the optimum jump of Experiment IV is a good place to start. At t = 100, the closest particle, k, to the new optimum location will now be at the new global best. Particle k will at this stage experience no acceleration, and therefore its next location is $\mathbf{x}_k(100) = \mathbf{x}_k(99) + w\mathbf{v}_k(99)$. Notice that the velocity $\mathbf{v}_k(99)$ is unlikely to be pointing towards $\mathbf{x}_{opt}(100)$. However, if $\mathbf{v}_k(99)$ has some component that lies along $\mathbf{x}_{opt}(100) - \mathbf{x}_k(99)$ then the new position $\mathbf{x}_k(100)$ will improve upon the previous position and may even be the new global best when the updates at this iteration are completed. Suppose this is so. Then, by a similar argument, $\mathbf{x}_k(101)$ will lie along the same trajectory $\mathbf{x}_k(100) - \mathbf{x}_k(99)$. Meanwhile the other particles, which were at their personal bests at t = 99, will experience accelerations towards $\mathbf{x}_k(99)$ of magnitude $c_1 r_1 |\mathbf{x}_k(99) - \mathbf{x}_i(99)|$. This will not be a large acceleration, but it will give the new velocity vector $\mathbf{v}_i(100)$, an additional component along the trajectory defined by $\mathbf{x}_k(100) - \mathbf{x}_k(99)$. At the next iteration, if the above scenario is played out, velocity components along the trajectory of particle k will again be reinforced.

Of course there may well be some jostling for leadership, but occasionally a leader will be found whose trajectory defines a line of global bests. The remaining particle velocities are pulled ever more in the direction of this trajectory and the accelerations place them ever closer to positions along this trajectory. The result is collinear motion along a line that is closing on $\mathbf{x}_{opt}$ but is not necessarily coincident with $\mathbf{x}_{opt}$. The leadership may then be exchanged, but motion along this line will always be reinforced until a final global best position is found which

is near to the point of closest approach between the trajectory – the 'end-point' - and $\mathbf{x}_{opt}$. Animations of many runs of the swarm provide empirical evidence that this scenario invariably occurs.

The stable trajectory is clearly seen in Figure 7. Figure 8 shows two snapshots that illustrate the attraction of the particles to the trajectory. At t = 114, particle 0 is displaced from the global best position. The acceleration is sufficient to place it very close to $\mathbf{x}_{gb}(114)$ at t = 115, but the global best has now moved along the stable trajectory to $\mathbf{x}_{gb}(115)$.

Consider now what happens when the global best is near to the end-point. Velocities perpendicular to the stable trajectory will be very small so that accelerations towards $\mathbf{x}_{opt}(100)$ will also be correspondently small. Moreover, the dissipative effects of the inertia weight will also be progressively slowing particle motion down. The result is that it may take a long time for the swarm to move away from the end of the stable trajectory, and the global best will hardly improve in the remaining iterations before the next optimum jump. The particle motion is now a spring-like oscillation along the stable trajectory, centered on the end-point. The stable trajectory and end-point are depicted in Figure 9 which shows the position of particle 0 and global best between iterations 110 and 140.

This analysis can be applied to the results of Experiment 1 (Figure 1). The plateau show that the global best scarcely improves over some tens of iterations. This is due to the collinear motion followed by oscillation about the end-point of the stable trajectory. In fact, there is a particularly bad run between iterations 2100 and 2500 when the particles never improve on their global best, which corresponds to an optimum value of 1000. This is 11 orders of magnitude from the best that the swarm is capable of finding (10^{-8}). It is these high values that push up the average best value found (Figure 2).

The charged swarm is less affected by this pathology since the collision avoiding acceleration will push particles away from the stable trajectory. In fact animations never show linear collapse; instead, the swarm maintains a near spherical shape, much more reminiscent of an insect swarm. However, this also has its drawbacks. Figure 3 does show some horizontal portions, for global bests in the range 10^{-1} to 1. The repulsions now work against exploitation so that better solutions than this are found in only 7 of the 50 optimum jumps.

The atomic swarm also does not suffer from the pathology of the neutral swarm. The charged particles allow for fast targeting, after which the neutral particles can continue searching the solution space in the near vicinity of the global best. Indeed, at the 100th iteration, the rate of improvement of best value is 11%, which shows that significant improvement can still occur. The corresponding rates for the neutral swarm and the charged swarm are 3% and 1%, indicating only slow progress is possible.

8 CONCLUSIONS

This work presents a new particle swarm algorithm based on an analogy of electrostatic energy. In addition, a dynamic search problem has been formulated that is more representative of real-world problems. The experiments considered here suggest that atomic particle swarms may offer a good strategy for dealing with such severe dynamic optimization over short time scales. Certainly this has been the case for the dynamic three dimensional parabolic function considered here, where the average best value obtained over 50 optimum jumps was, by the 100th iteration, 1.12×10^{-4}. This compares very well with the equivalent figure of 125 for the conventional PSO algorithm.

The poor behavior of the conventional (i.e., neutral) particle swarm seems to be due to a curious pathology of 'linear collapse', just after problem optimum jump. This was observed in animations and analysis suggests that the cause is the establishment of a linear trajectory that links global best positions and serves as an attractor for the swarm. At the end of this stable trajectory is a stable global best position, which can be some way from the optimum location, and from which the swarm has difficulty improving upon.

The charged particle swarm has the advantage that the particle trajectories are always around an extended swarm shape, allowing good global search. The maintenance of an extended swarm was the reason for the use of collision avoidance in the earlier work on improvised music (Blackwell and Bentley 2002a,b). In this context, it is desirable to have a very fast swarm response to a changing audio input, yet undesirable for the swarm to cluster too closely around a target – this would lead to dull melodies and parody. The downside is that the particle repulsions prevent detailed exploration of the search space.

However, experiments suggest that a swarm of neutral and charged particles (reminiscent to representations of the atom) does not suffer from linear collapse, and always allows for detailed exploitation. The advantage of an atomic swarm over randomizing strategies (e.g., where the particle positions are randomized when an problem optimum shift is noticed) is one of simplicity. No further analysis is needed to tell just when a change has occurred, and how the swarm should respond to this change.

References

P.J.Angeline (1997). Tracking extrema in dynamic environments. *Proc. Evolutionary Programming VI*, Indianapolis, IN. Berlin: Springer-Verlag (Berlin), pp 335-345.

T. Bäck (1998). On the behavior of evolutionary algorithms in dynamic environments. *Proc. Int. Conf. on Evolutionary Computation*, Anchorage, AK. Piscataway, NJ: IEEE Press pp 446-451

Blackwell, T. (2001) "Making Music with Swarms," *MSc thesis, University College London.*

Blackwell, T. and Bentley, P. J. (2002a) Improvised Music with Swarms, *Proc CEC 2002* (to be published).

Blackwell, T. and Bentley, P. J. (2002b) Don't push me! Collision-Avoiding Swarms, *Proc CEC 2002* (to be published).

M.Clerc (1999). The swarm and the queen: towards a deterministic and adaptive particle swarm optimization. *Proc CEC 1999*. Washington, DC, pp 1951 – 1957.

R.C.Eberhart and Y.Shi (2001a). Particle swarm optimization: Developments, applications and resources. *Proc CEC 2001*. Piscataway, NJ, pp 81-86.

R.C.Eberhart and Y.Shi (2001b). Tracking and optimizing dynamic systems with particle swarms. *Proc CEC 2001*. Piscataway, NJ, pp 94-100.