Kernel methods for PSOs

W. B. Langdon Riccardo Poli
Christopher R Stephens

Department of Computer Science, University of Essex, UK

Technical Report CSM-443

20 December 2005

Abstract

We report initial experiments on evolving kernel functions which describe the average behaviour of a swarm of particles as if it was responding as a single point moving on a landscape transformed by the kernel. Such kernels may help explain swarm and population approaches leading to extended population systems, XPS. The swarm of particles are from a simple particle swarm optimiser solving one dimensional multi-modal 3 peaks and Rastrigin problems. The standard Java genetic programming implementation TinyGP is used.

1 Group Computation

In plant and animals systems it is the rule rather than the exception that individuals, at least for some critical periods, live within groups. Why is this? There are many things that a group of individuals can do that an isolated individual cannot. E.g. gather rain drops, overshadow your competitors, build cathedrals. We shall concentrate upon "computation"; what can groups (e.g. swarms, shoals) of individuals do better (or indeed worse) than the individuals themselves? Ours is a theoretical, computer science, approach, in which the individuals are members of small standard particle swarm optimiser (PSO). However the idea of the collective "perceiving" its environment through a mathematical "kernel" mapping the individual's senses, is widely applicable.

These initial experiments are aimed at showing the kernel approach has at least some credibility. The idea is to see if we can model movement of a PSO swarm, composed of individuals, each moving individually, as a whole, which responds to the underlying optimisation problem via a perceptive field.

We track a real PSO swarm. Keep track of how individuals move on some simple landscapes. Can we explain this, as movement of the whole swarm (e.g. its centre of mass) which "perceives" the fitness landscape via a receptive field. The receptive field might be centred at the centre of the swarm. We implement the receptive field as a filter (convolution). Use genetic programming to find suitable kernel. The simulated whole swarm uses hill climbing on landscape it perceives via the convolution of the filter and the real problem.

In a school individual fish have limited perception. This is dominated by the movement of other members of the school adjacent to them. However they also have some more long range sensory information, e.g. the my smell a "danger signal" released when a fish is attacked by a predator.

The following three sections describe briefly the particle swarm optimiser, the hill climbing strategy used on the transformed problem space and the genetic programming system used to evolve the kernels. Section 5 describes the experiments on the PSO moving on the three peaks landscape and on the Rastrigin problem. Sections 6 and 7 discuss our results and conclusions.

2 PSO

The five member PSO swarm is very simple. There is no constriction, instead speed is limited to 1.0. The PSO is run for 25 cycles.

The first five cycles are treated as setting time and discarded. I.e., the GP tries to make the hill climber match only the last 20 generations of the PSO run.

3 Hill Climber

The hill climber is started at the centre of mass of the PSO swarm. It next twenty moves are compared with the centre of the PSO at successive generation. The hill climber response directly to the local gradient ∇, accepting all moves, regardless of their fitness. The learning rate ϵ is one. The local gradient is, of course, given by convolving the actual landscape with kernel gradient evolved by GP. For simplicity the function evolved by GP is treated as the gradient of the kernel, rather than the actual kernel. We do not need the actual kernel during the run. Should it be needed, it can be constructed by integrating its differential after the run.

4 GP

We use the state of the art TinyGP genetic programming system, cf. Table 1. It evolves a kernel, i.e. a function, which is convolved with the one-dimensional problem landscape which is to be optimised. This gives another landscape which is "perceived" by a simple hill climber $x_{t+1} = x_t + \epsilon \nabla x$. Where ∇x is the gradient and ϵ is the learning rate. The goal for the GP is to evolve a kernel which causes the hill climber to move so as to resemble movement of the whole PSO swarm.

The hill climber is deliberately simple in these initial experiments. It has fixed step size and learning rate.

4.1 GP Fitness function

The PSO is run 5 times. Fitness is the sum over the last twenty generation of each run of the squared distance between the position of the hill climber and the mean position of the members of the swarm.

5 Results

5.1 3 Gaussians

The first test landscape is asymmetric and multi-modal. It is created by adding together three Gaussian curves, cf. Figure 1. The behaviour of the five particle PSO on it, and the kernel predictions are given in Figures 2 and 3. Figure 2 shows in all runs, the swarms are able to climb upto the first peak but are not trapped by it. Instead they all get at least as far as the

Table 1: TinyGP PSO Kernel Parameters

Function set:	$+ - \times$ DIV[a]
Terminal set:	110 terminals, including: The remaining terminals are constants uniformly randomly chosen in the range $0 \ldots +1$
Fitness:	See Section 4.1
Selection:	steady state binary tournaments for both parent selection and who to remove from the population
Initial pop:	Trees randomly grown with max depth of 4 (root=0)
Parameters:	Population 1000. 10% crossover, 90% mutation (2% chance of mutation per tree node).
Termination:	generation 100

[a] DIV is protected division I.e. if $|y| <= 0.001$ $\text{DIV}(x, y) = x$ else $\text{DIV}(x, y) = x/y$.

second peak. Here the swarms diverge; two succeed in passing over it and start oscillating about the highest fitness value. Whilst in the remaining three runs, the swarm oscillates about the intermediate peak. Note this divergence does not prevent the Hill climber, seeing the PSO's landscape via its kernel (evolved by genetic programming) and successfully following the centre of the swarm in each run.

5.2 Rastrigin

The Rastrigin benchmark is widely used since its many local peaks give interesting dynamics. See Figure 4.

Again genetic programming is able to automatically produce a kernel which transforms the landscape seen by the five members of the PSO swarm into a landscape where a simple hill climber emulates their centre of mass. See Figures 5 and 6.

6 Discussion or Future Work

Our motivation has been to find examples where the swarm behaves as a unit. More particularly where this unit "perceives" its environment (i.e. the artificial landscapes the PSO was run on) via a "kernel". Each member of the swarm provides very local information. The kernel combines the sensory information from each of the individuals into something which guides the unit as a whole. The idea with the hill-climber is just to give something simple to represent the response of the idealised unit-group-swarm.

The PSO is deliberately a very simple velocity limited PSO. There is no constriction etc. Two things may help it avoid getting stuck at the top of a local hill: 1) Each particle's momentum tends to carry it past local optima; 2) Another member of the swarm may be on a better hill and so attract the rest of the swarm towards it.

Kernels provide a mathematical way of combining measurements taken at different locations (and/or at different times). Kernels are often used in computer vision and signal processing but often a great deal of care and time is needed to design them, so people try and use one of the existing standard kernels. Genetic programming provides an automatic means of generating kernels to suit each circumstance.

With loosely coupled individuals, like fish in a shoal, we do not belive that they act via a

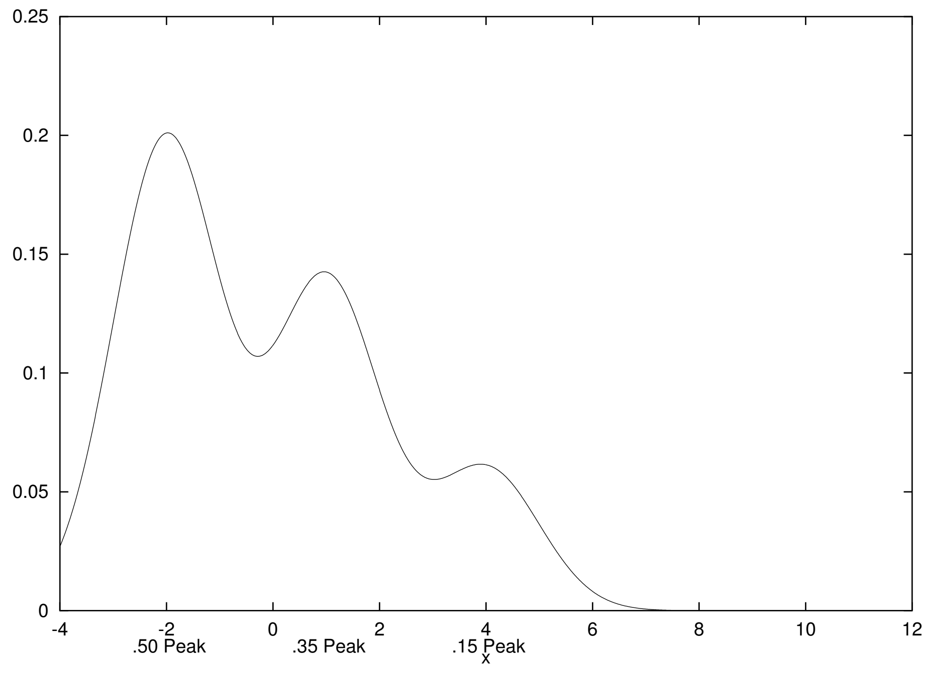

Figure 1: Three Gaussians test problem.

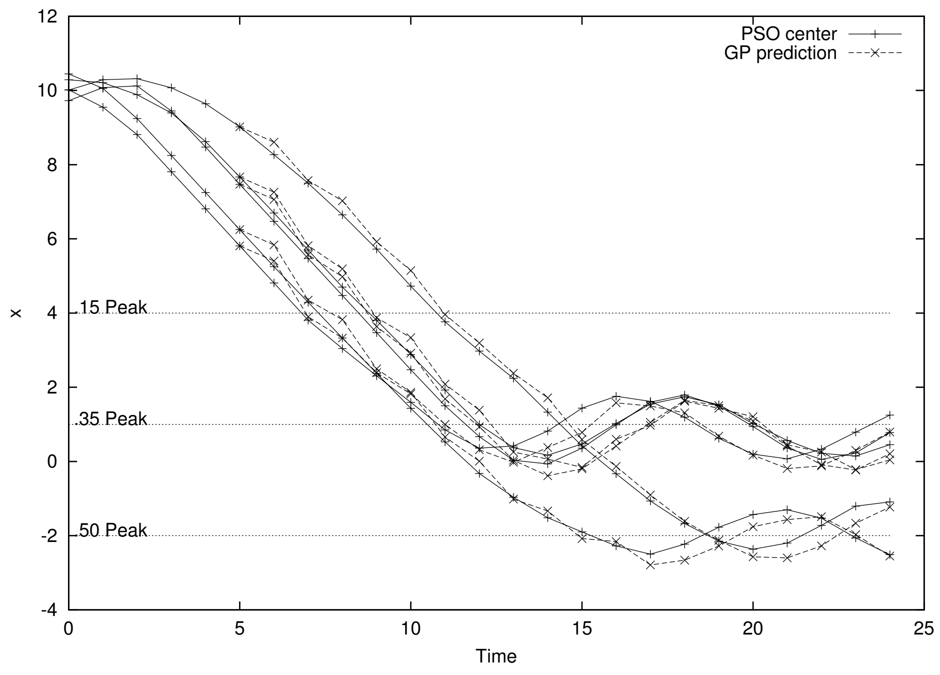

Figure 2: Testing evolved kernel. Comparison of five actual PSO runs (solid lines) with five hill climbing runs on three Gaussians landscape (dashed lines). Training RMS error 0.27 v. test RMS error 0.32

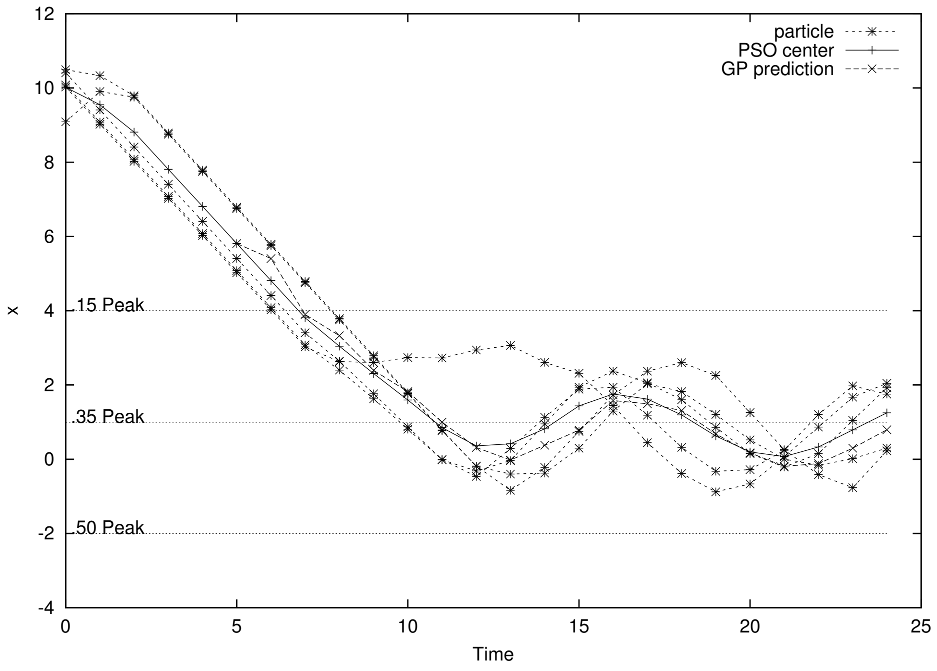

Figure 3: First PSO run from Figure 2 showing location of individuals within the PSO swarm.

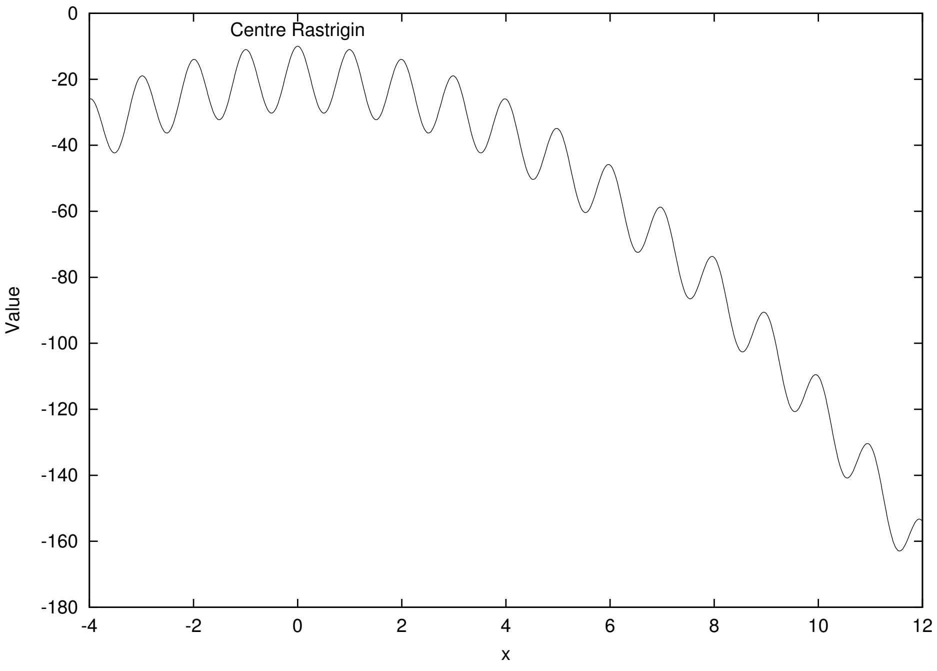

Figure 4: The one Dimensional Rastrigin benchmark problem.

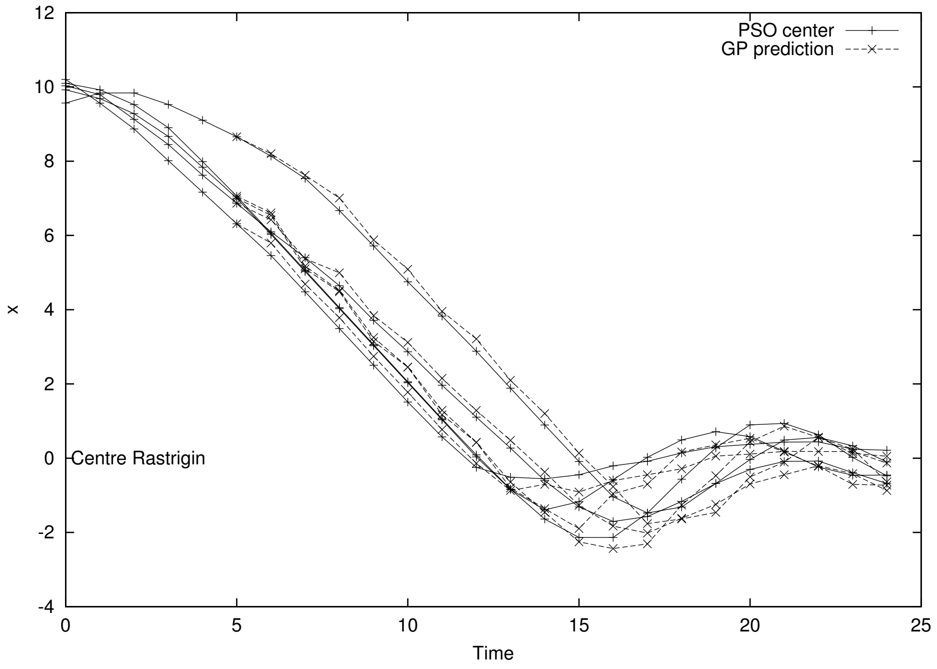

Figure 5: Testing evolved kernel on Comparison of five actual PSO runs (solid lines) with five hill climbing runs on one Dimensional Rastrigin. Training RMS error 0.28 v. test error 0.33

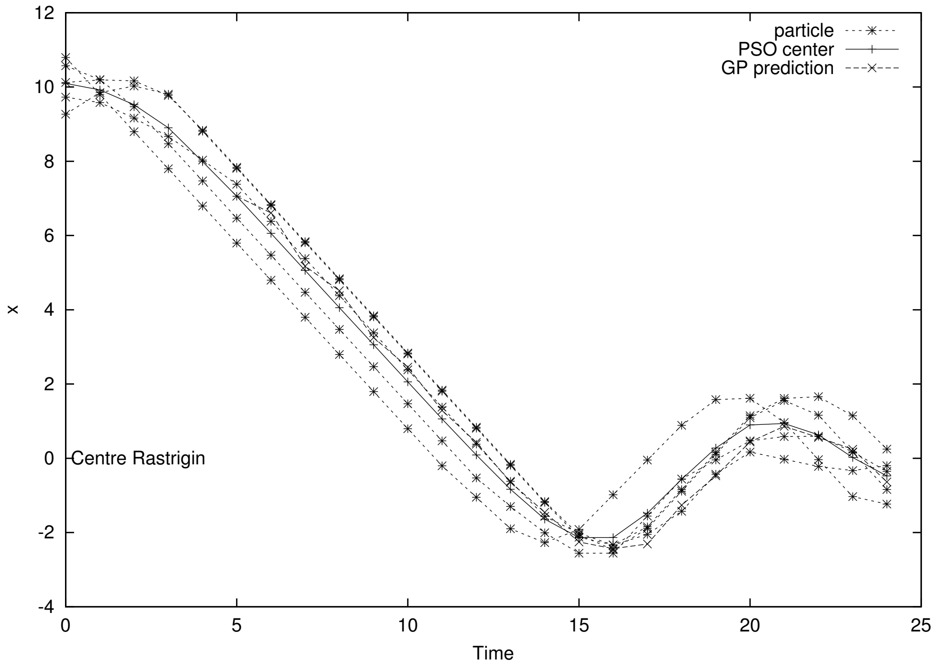

Figure 6: First test run of evolved kernel on one Dimensional Rastrigin, cf. Figure 5.

centralised kernel, however a kernel may be a good way of describing how the group behaves. With more integrated systems (e.g. neurons in a retina) the kernel idea may be even better.

Experiments show some evidence of PSO “thinking” as a whole? This is obviously only initial work. It needs to be expanded greatly. Can we use kernel approach to explain how the collective makes decisions in uncertain environments? Does it help show/explain/quantify emergence of behaviour? Can we treat the (human?) Brain as a collective of neurons?

The kernel approach can also be applied in time domain. That is, in the same way that spatial kernel can model a group an individual spread over some area of space, a temporal kernel can show the effects of an individual “knowing” sensory readings from its past locations. i.e. to show benefits/drawbacks of memory. One could even think of combined space and time kernels, showing how a collective uses “group memory”.

7 Conclusions

We have shown on two problems that genetic programming can automatically find simple kernels which transform the optimisation problem seen by individual members of a PSO swarm into a single landscape on which a single point matches the average behaviour of the whole swarm. Given the stochastic nature of PSOs, the agreement between the movement of the single point and the PSO centre of mass can be surprisingly good.

Acknowledgements

Support by EPSRC grant GR/T11234/01.