IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661, p- ISSN: 2278-8727Volume 14, Issue 4 (Sep. - Oct. 2013), PP 44-48
www.iosrjournals.org

A Reflective Swarm Intelligence Algorithm

Blamah Nachamada Vachaku
Department of Computer Science, University of Jos, Jos, Nigeria;

Abstract: *Swarm Intelligence (SI) algorithms are heuristics for finding the optimal solutions of optimization problems. They are made up of groups of swarms that interact with one another in the search effort within their environment. A reflective SI algorithm is presented, where members of the swarm are able to reflect backward to reconsider historic actions in order to adjust their search behaviors and stick to better results, which make the algorithm to perform robustly.*
Keywords: *Swarm intelligence, heuristic, retrospective*

I. Introduction

Swarm Intelligence (SI) is inspired by the social hunting nature of animals such as the flocking behaviors of birds moving towards an optimal goal, and it is in the class of technique based around the study of collective behavior in decentralized, self-organized systems.Each particle is subject to a movement in a multidimensional space. The particles usually have memories, thus retaining part of their previous states, and there is no restriction for particles to share the same point inspace, but in any case their individuality is preserved (Brownlee, 2011; Gupta, Sharma and Singh, 2012).

As a case of SI, we will look at the Particle Swarm Optimization (PSO), which was first developed by Kennedy and Eberhart (1995) as an optimization method for continuous functions. The major idea about this scheme is that particlesmove towards more suitable members of the swarm and generally bias their movements towards historically good areas of the environment and they try to achieve the optimal goal by cooperating with their neighbours in addition to taking independent decisions and actions.

II. Reflective Swarm Intelligence

The particle swarm intelligence (PSO) algorithm is a population-based algorithm, where a set of potential solutions evolves to approach a convenient solution for a problem, and the aim is to find the global optimum of the fitness function defined in a given search space. The particles that are part of a society hold an opinion that is part of a "belief space" (the search space) shared by every possible particle. Particles may modify this "opinion state" based on either the knowledge of the environment (its fitness value), the individual's previous history of states (its memory), or the previous history of states of the individual's neighbourhood (Confort and Meng, 2008).

Following certain rules of interaction, the individuals in the population adapt their scheme of belief to the ones that are more successful among their social network. Over the time, a culture arises, in which the individuals hold opinions that are closely related.

Intelligent reasoning is connected with the way *reasoning* is carried out in order to arrive at a conclusion.It refers to the ability to come to correct conclusions about what is true or real, and about how to solve problems(Davidson, 1992).Reasoning in a general sense is a broad subject matter that refers to the capacity to make sense of things, to establish and verify facts, and to change or justify practices and *beliefs*(Kompridis, 2000).

Aparticle within the swarm will arrive at an alternative by first deliberating on the available options, and then make decisions by acting on the best alternatives. A Particle starts by having a particular set of beliefs whichare stored in a belief database, and then commits to intentions for actions based on the initial beliefs and desires. As time progresses, the beliefs about the swarm may be refined, and the particle's desires and intentions may also be redefined to reflect the changes in the belief database.

After aparticle deliberates and produces intentions to which it is committed, it needs to plan how to accomplish the intentions based on the current state of the environment and the actions that are available to it.
So a particleperceives its environment and then adjustsits belief database appropriately, upon which it derives its intentions, and then reasons on how to take an action that alters the swarm, which advances the system towards a potential solution.

III. Particle Swarm Optimization Behavior

In this section, we will look at the behaviors of particles in a typical swarm. In our design, we willmodel the appropriate qualities of the conventional PSOand then introduce concepts that improve on the overall systems performance as an optimization technique.

The particles within the system consider problems by weighing conflicting considerations for and against competing options, where the relevant considerations are provided by what the particle desires or values and what the it believes (Bratman, 1990; Wooldridge, 2009).

Aparticle takes action by first deliberating on *what* state of affairs to achieve from the available options, which represents its *Intentions* that alter its state of mind. Then the particle reasons on *how* to achieve the chosen state of affairs, which results in a plan of how best to achieve the option. By so doing, intelligence is built into the system.

The following specific issues are considered in our model:

1. The concept of neighbourhood as a more sophisticated information-sharing scheme among particles was introduced because the inertia weight PSO model gets trapped easily in local minima, especially in complex problems, where the swarm easily collapses due to complete diversity loss (Parsopoulos and Vrahatis, 2010).

The main idea behind the neighbourhood approach was the reduction of the global information exchange scheme to a local one, where information is diffused only in small parts of the swarm at each iteration. Each particle assumes a set of other particles to be its neighbours and, at each iteration, communicates its best position *only* to these particlesinstead of to the whole swarm. Thus, information regarding the overall best position is initially communicated only to the neighbourhood of the best particle, and successively to the rest through their neighbours.

With this approach,a particle is tied to a fixed neighbourhood for interaction with each iteration without *planning* ahead and envisaging better fitness values with other neighbourhoods within the same iteration; so when aparticle belongs to a neighbourhood, it does not directly share neighbourhood best information with other particles outside its immediate neighbourhood within a single iteration to see if such interaction will yield better fitness values.

In our approach to PSO, cohesionand diversity within the search space is increased in order to avoid*blind commitment* that easily gets particles trapped in specific local minima. The particles dynamically alternate neighbours (whatever neighbourhood topologyis used) in the search process. In each iteration, aparticle computesseveral fitness values in parallel (based on neighbourhood bests of the main neighbourhood, and other neighbourhoods), and keepsthe history in the belief database for future reference. Depending on the best results obtained with time, the particlesticks to neighbours that yield better fitness values. So the best global behaviour emerges as the particles interact.

2.Learning is highly desirable within a complex system like PSO.A belief database will be designed, which is well suited for keeping the particles' learned experiences overtimeas the particles keep refining their beliefs.

The qualities described here make the particles more autonomous and intelligent.

IV. Reflective Entities

In this section, we present the different entities that collectively describe the entire system. The Particle Swarm Optimization (PSO) model is presented using the velocity and position update equations that follow:

$$V_i^{t+1} = \omega V_i^t + C_1 R_1 (pBest_i^t - X_i^t) + C_2 R_2 (lBest_i^t - X_i^t) \quad (1)$$

$$X_i^{t+1} = X_i^t + V_i^{t+1} \quad (2)$$

where$t$ denotes the iteration counter, R_1 and R_2 are random numbers distributed within [0,1], C_1 and C_2 are weighting factors representing the cognitive and social parameters respectively, $pBest_i^t$ is the best position known so far by the particle i, $lBest_i^t$ is the neighbourhood best, V_i^{t+1} is the velocity of particle i initeration$t+1$, and X_i^{t+1} is the particle's position in iteration $t+1$.

Variant of the PSO (Parpinelli and Lopes, 2011) shows that population can grow or shrink, which means in reality, the particles in a swarm, within each iteration, can communicate with any number of particles ranging from 1 to $n-1$, n being the swarm population.For this reason, we build the system such thateach particle within thesystemis able to communicate to various neighbourhoods, and does not limit its communicationto a fixed number of particlesall the time. So each particle is assigned few other prospective neighbourhoods besides the main neighbourhood it belongs to.In each iteration, the $lBest_i^t$values of these prospective neighbourhoods are communicated to the particle alongside the $lBest_i^t$ value from the main neighbourhood, and the particle computesboth prospective and main *velocities, positions*and *fitness* values in parallel. After the computations, the particle stores all the prospective values in a belief database, but uses the main fitness value to keep membership of the main neighbourhood by moving in its direction. After a certain time-stampequivalent to the size of the belief database, the particle takes appraisal of its execution history and compares them with the values in the belief database.

The neighbourhoods of the system is then reformulated based on more promising fitness values of particles up to the time of appraisal, and the particles will move in the directions of best fitness values. The time stamp is reinitialized and the process is repeated.
Aparticle chooses neighbourhood based on its previous knowledge and experience, with its desire towards more promising neighboursas it updates its beliefsand subsequently gets attracted towards more promising regions.
Prior to the algorithm, we will present some descriptions that will aid us describe the entire system.

4.1 State of Environment

The state of environment in which the particle's search space may be,is defined as follows:

$$E = \{P_1, P_2, . . . \ P_N\} \quad (3)$$

where$P_i = (P_{i1}, P_{i2}, . . . \ P_{in})^T$isthe best positions ever visited by each particle, representing the present state ($i = 1, 2, \ldots, N$, N being the population size and n the current iteration counter).
We are going to describe more concepts in order to aid us to define anparticlet's action.
Each particle has a range of actions at its disposal, which are the consequences of the particle's invocation.In order to accommodate more neighbourhoods and prospective positions, we slightly modify the velocity and position formulae of equations (1) and (2) by simply converting them to $k -$ dimensional vectors.

$$V_i^{k(t+1)} = \omega V_i^{k(t)} + C_1 R_1 \left(pBest_i^{(t)} - X_i^{k(t)}\right) + C_2 R_2 (lBest_i^{k(t)} - X_i^{k(t)}) \quad (4)$$

$$X_i^{k(t+1)} = X_i^{k(t)} + V_i^{k(t+1)} \quad (5)$$

$$1 \leq k \leq \eta$$

Each particle uses only one particle's best $pBest$ within each iteration, but the velocity V, neighbourhood best $lBest$, and position X, all become k-dimensional vectors, where k ranges between 1 and the total number of neighbourhoods ηin the system.
So in practical implementation, the value of k depends on the number of neighbourhoods each particle takes concurrently while executing. If $k = 1$, it becomes the conventional PSO; if $k = 2$, it means each particle considers 2 neighbourhoods – its present neighbourhood and 1 prospective neighbourhood; if $k = 3$, then each particle considers 3 neighbourhoods – its present neighbourhood and 2 other prospective neighbourhoods, and so on.
Let the actions in equations (4) and (5) be represented by the set

$$V_x = \{V, X\} \quad (6)$$

We represent subtractions, additions and multiplications as basic operations, which we represent as:

$$B_o = \{Sub, Mul, Add\} \quad (7)$$

4.2 Action

If we generally define the set of actions at the disposal of particles as$Ac = \{\alpha, \alpha', \ldots\}$, then specifically, the i^{th} particle in the system has these actions:

$$Ac_i = \{V_x, C, \sigma, B_o\} \quad (8)$$

where$V_x$is the set of velocity and position functions described by equation (6), C is a communicator (Blamah, Adewumi and Olusanya, 2013)which the particle uses to communicate with other particles, σ is a recap function that permits aparticle to appraise its history from the last time stamp in order to decide whether or not to change neighbourhoodwithin the set $\mathcal{G}$ (explained in equation (12)).
Particles in the search space have single-minded commitments, because a particle continues to maintain an intension of improving the fitness values within a particular neighbourhood until it believes either that the intension has been achieved, or else it is no longer a more feasible option to remain in that neighbourhood, in which case it is rational for the particle to move away to a more promising neighbourhood.
In our model, we assume that the size of neighbourhoodscan vary because there may beincrease or decrease in population;particles can move from one neighbourhood to another, or any other factor mayoccur that can alter the population size (Obagduwa, 2012).

4.3 Runs

A run, r, of a particle in an environment is a sequence of interleaved environment states and actions. If we let R be the set of all such runs, then we have:

$$R = \{r, r', \ldots\} \quad (9)$$

LetR^{Ac} be the subset of these that end with an action and R^Ebe the subset of these that end with an environment state.

4.4 Transformation

When a particle invokes an action on an environment, it transforms the environment state, and this effect is modeled by the state transformer function defined as:

$$\tau : R^{Ac} \rightarrow 2^{E} \tag{10}$$

where2^{E} is the power set of E.

This means that from runs which end with actions taken by a particle, the system will always end up in a particular environment state; taking an action by a particle on a previous environment state moves the environment to another state.

4.5 The Particles

In a swarm, the particles drive the system. The state of the environment emerges as a result of the particles' actions – basedon the behaviours and interactions among the particles. Since actions are produced by particles when they execute in the system, we model particles as function of execution, which yield an action (whose effect is the state transformer function). Thus, a particle is defined as:

$$A : R^{E} \rightarrow Ac \tag{11}$$

So if an action, say the position update function $X \epsilon Ac$, is desired of a particle A', theparticle produces this action by executing on an existing run ending with environment state, say r', which is its current position, as follows:

$$X = A'(r')$$

This leaves the run to end with an action. The effect of taking this action, which is modeled by the state transformer function,τ, is to produce a new environment state.

4.6 Environment

We define an environment as a tuple:

$$\xi = \langle E, e_0, \tau, T, \mathcal{G} \rangle \tag{12}$$

where$E$ is the set of environment states described by equation (3), $e_0 \in E$ is an initial state, τ is a state transformer function described by equation (10), T is the active topology in the environment,and $\mathcal{G} = \{g_1, g_2, \ldots, g_\eta\}$ is a set of η neighbourhoods such that:

$\emptyset \notin \mathcal{G}$,

$\cup \mathcal{G} = S$, S being the swarm, and

$\forall g_i, \forall g_j \left(g_i \in \mathcal{G} \wedge g_j \in \mathcal{G} \wedge g_i \neq g_j\right) \Rightarrow g_i \cap g_j = \emptyset,\ 1 \leq i,j \leq \eta.$

4.7 Swarm

The Swarm $S$is defined to be the set of all particles,as follows:

$$S = \{A_1, A_2, \ldots, A_n\} \tag{13}$$

where$\cup \mathcal{G} = S$. The Swarm System is thus defined as $R(S, \xi)$, where R is the set of all runs.

V. Algorithm

Particles need to *plan*ahead and envisage better fitness values with other neighbourhoods within the same iteration by sharing information with particles outside the main neighbourhood.The particles thus get committed to achieving better fitness values by reasoning and dynamically alternating neighbours in the search process.

The initial intentions of each particle are predefined such that each particle is committed to its present neighbourhood. But with time, each particle keepsto the prospective neighbourhood which yields a better fitness value after the elapse of the chosen time stamp. This will eventually yield the global best of the fitness function.

With the initial beliefs and the initial intentions, a particle is executed using algorithm 1.

Algorithm for Reflective Swarm Intelligence

Randomly initialize the whole swarm
Initialize $timeStamp = 0$; $threshold = thresholdVal$;
While (termination criteria is not met) {
    if$(timeStamp = threshold)$ {
        for $(i = 0;\ i < swarmsize;\ i++)$
            $Execute(plan)$;
        $timeStamp = 0$;
    }
    for $(i = 0;\ i < swarmsize;\ i++)$ {
        $Evaluate\ f(x_i)$;//where x_i is the position of particle i
        if$(f(x_i) > f(pbest_i))\ pbest_i = x_i$;
        if$(f(x_i) > f(lbest))\ lbest = x_i$;
    }
    for $(i = 0;\ i < swarmsize;\ i++)$ {

```
15:            calculate (v_i);
16:            update (x_i);
17:      }
18:      timeStamp ++;
19: }
```

In addition to the basic particle swarm optimization (PSO) processes, this algorithm handles the choice of prospective neighbourhood between lines 4 and 8, and line 18 increments the timestamp.

The overall behaviour of the particles within the system isrepresented as shown in Fig. 1. The un-shaded region with broken line represents all searches made by particles within neighbourhood g_1, and the un-shaded region with solid line represents searches made within neighbourhood g_2. The intersection of $g_1$and g_2 represents where the fitness values from the two regions coincided, and the searches made by A_{12} for both fitness f' and f'' span the two regions, which is indicated by the dot-shaded region with double solid line. A switch from g_1 to g_2 gave a better fitness due to the particle's ability to reflect back to previous behaviours and stick to historically better results.

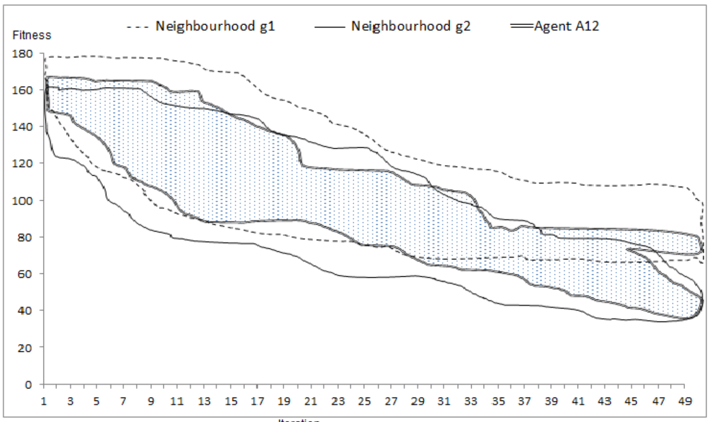

Figure 1: Fitness values within Neighbourhoods$g_1$and g_2 for all particles.

VI. Conclusion And Future Work

Learning capabilities were built into the particles to dynamically adjust their optimality behaviours, and autonomy was achieved by the use of communicators that separate anparticle's individual operation from that of the swarm.The traditional PSO algorithm was modified in order to incorporate the desired learning capabilities into the particles. The particles were designed to make parallel computations of fitness values based on the $lbest$ of manyneighbourhoodsand store them in the belief database for future reference, and retrospectively stick to historically better fitness values.As part of our future work, we intend to design the scheme to work as a complete multi-agent system and apply it to a classical optimization model.

References

[1] Parpinelli, R.S., and Lopes, H.S. (2011). New inspirations in swarm intelligence: a survey, *International Journal of Bio-Inspired Computation*, 3(1), pp.1–16.

[2] Bratman, M.E. (1990).*What is Intention?* In Cohen, P.R., Morgan, J.L., and Pollack, M.E., editors, Intentions in Communication, The MIT Press: Cambridge, MA.

[3] Brownlee, J. (2011).*Clever Algorithms: Nature-Inspired Programming Recipes*. 1st ed., Lulu. http://www.CleverAlgorithms.com

[4] Parsopoulos, K.E., and Vrahatis, M.N. (2010). *Particle Swarm Optimization and Intelligence: Advances and Applications*, Information Science Reference, Hershey, (NY).

[5] Confort, M and Meng, Y. (2008). Reinforcing Learning for Neural Networks using Swarm Intelligence, *IEEE Swarm Intelligence Symposium*, St. Louis MO, USA, Sept., 21-23, 2008.

[6] Davidson, H. (1992). *Alfarabi, Avicenna, and Averroes, on Intellect*, Oxford University Press, pp.6.

[7] Gupta, P., Sharma, K., and Singh, P. (2012). A Review of Object Tracking using Particle Swarm Optimizatio, *VSRD, International Journal of Electrical Electronics & Comm. Engg. 2(7).*

[8] Blamah, N. V., Adewumi, A. O. and Olusanya, M. O. (2013). A Secured Agent-Based Framework for Data Warehouse Management. *Proceedings of IEEE International Conference on Industrial Technology (ICIT)*, pp1840-1845, Cape Town.

[9] Kennedy, J., and Eberhart, R. (1995). Particle Swarm Optimisation. In: Proceedings of the IEEE Conf. on Neural Networks, Perth.

[10] Wooldridge, M. (2009), *An Introduction to MultiAgent Systems*. 2nd ed., John Wiley & Sons Ltd, Chichester.

[11] Kompridis, N. (2000). So We Need Something Else for Reason to Mean, *International Journal of Philosophical Studies*,8(3), pp.271-295.

[12] Obagduwa, I.C. (2012). Cockroaches Optimization Algorithms, Seminar paper presented at the Progress Seminar, March 3rd, UKZN, Durban.