International Journal of Soft Computing 3 (4): 297-301, 2008
ISSN: 1816-9503

A Novel Adaptive Life Cycle Model: Combining Particle Swarm Optimization and Memetic Algorithms

[1]P. Jaganathana and [2]K. Thangavel
[1]Department of Computer Science and Applications, Gandhigram Rural University, Gandhigram-624 302, Tamil Nadu, India
[2]Department of Computer Science, Periyar University, Salem-636011, Tamil Nadu, India

Abstract: Effective discovery of classification rules for the high dimensional data is becoming one of the hard search problems and hot research area. Heuristic search algorithms provide an approximate solution to hard search problems within the reasonable time. Inspired by the biological life cycle of nature, we introduce a Novel Adaptive Life Cycle Model (NALCM) which applies both Memetic Algorithms (MAs) and Particle Swarm Optimization (PSO) to create a well-performing hybrid heuristics for the discovery of rules. In the proposed model, candidate solutions are represented as individuals and based on the fitness, they can decide to become either a MA individual, a particle of a PSO. Results are compared with other search algorithms such as Particle Swarm Optimization and Genetic Algorithms. The proposed model achieves better performance.

Key words: Hybrid heuristics, life cycle model, particle swarm optimization and genetic algorithms

INTRODUCTION

In biology, the term life cycle refers to the various phases an individual passes through from birth to death (Lawrence, 1996) often leads to drastic transformations of the individual with stage specific adaptations to a particular environment. This phenomenon is particularly amazing in considering the genome remains the same within each cell and life stage, whereas the morphology and behaviour of the phenotype can change drastically in accordance to the requirements of the life stage niche. Some life cycle changes in nature are one-time events such as sexual maturity. Other changes are re-occurring, such as mating seasons. These stages are genetically determined and the individuals have little or no influence on the change of the life cycle stage. The transitions between life cycle changes are often triggered by environmental factors. Environmental changes often determine transitions from one life cycle stage to another. Some animals are able to sense and predict these changes and can actively decide to alter their life cycle stage. The inspiration of this study is obtained from the ability of an individual to actively decide about its kind of life form in response to its success in its current environment and the Darwin's evolution theory.

The main idea behind the Novel Adaptive Life Cycle Model is to create a self-adaptive search heuristic in which each individual (containing the candidate solution) can decide whether it would prefer to belong to a population of a Memetic Algorithm or a Particle Swarm Optimization. The decision of the individual depends on its success in searching the fitness landscape. The motivation for this hybrid approach was that each of these search techniques on its own has its specific problem dependent strengths and weaknesses. MAs, for instance, are widely applicable and particularly powerful when domain knowledge can be incorporated in the operator design and high probability to optimum solution. However, PSO (Kennedy and Eberhart, 1995) can achieve clearly superior results in many instances of numerical optimization, but there is no general superiority compared to GAs (Kennedy, 1999; Shi and Eberhart, 1998; Suganthan, 1999; Angeline, 1998; Lovbjerg *et al*., 2001). Local search algorithms such as tabu Search and hill-climbing are good for local search with a high probability of finding the closest optimum. However, for multimodal functions, their performance is highly dependent on their starting position and hill-climbing techniques often convergence prematurely at local optima. Their main weakness compared to population based approaches, such as GAs and PSOs, is that candidate solutions neither compete nor cooperate (Michalewicz and Fogel, 2000). The main goal of the proposed Novel Adaptive Life Cycle Model is to make

Corresponding Author: P. Jaganathana, Department of Computer Applications, PSNA College of Engineering and Technology, Dindigul-624 622, Tamil Nadu, India

a self-adaptive hybrid heuristic approach towards a problem invariant search technique that can further take advantage of the changing search requirements during the optimization, such as initial exploration and local fine-tuning towards the end of the run.

METAHEURISTICS MODEL

Metaheuristics have been used to construct efficient models which are capable of finding high-quality solutions for many optimization problems with large search spaces. This is applied to the process of rule discovery from the large amount of data as it is being considered as a hard search problem. The classification rules are the combination of logical conditions and the prediction classes. The above fact has motivated researchers to apply heuristic models from evolutionary algorithms and swarm intelligence to classification rule discovery (Parpinelli *et al.*, 2002).

Each of the metaheuristic, has relative strengths and weaknesses. For example, population based methods are better in exploration of the search space but intensification can be more effective in tabu search. Over the years, researchers have developed new algorithms by hybridizing different metaheuristics in order to get a combination of the benefits of their relative strengths. Talbi (2002) who proposed taxonomy of the hybrid metaheuristics, provided the classification (Fig.1) based on algorithm design. In a low level hybridization, a particular feature of a metaheuristic is replaced by a feature from another metaheuristic. In a high level hybrid model, each individual operates in a self contained manner. Relay versus teamwork distinctions address the type of interaction between the metaheuristics. In a relay hybrid model, individual heuristics run one after another by using output from previous one. In the case of teamwork hybridization, parallel agents performing different metaheuristics run at the same time in the search space.

The life cycle model: The Life Cycle Model proposed by Krink and Lovbjerg (2002) consists of individuals starts with PSO particles, which can turn into MA individuals, then back to particles and so on. The structure of the Life Cycle model is illustrated in Fig. 1. In all these heuristics, one fitness evaluation per individual per iteration is used. A Life Cycle individual switches its stage when it has made no fitness improvement for more than 50 iterations.

Adaptive life cycle model: The Novel Adaptive Life Cycle Model applied to classification rule discovery in this study is inspired from the Life Cycle model developed by Krink and Lovbjerg (2002). Based on the classification provided by Talbi (2002) above, the proposed life cycle model is a high-level, relay hybrid of particle swarm optimization and memetic algorithms. The model uses a simple self adaptive transition relay method between heuristics in order to improve performance as outlined in Fig. 2. It has been applied to various combinatorial optimization problems and delivered better results than non-hybrid single methods in some numerical optimization problems (Krink and Lovbjerg, 2002). The proposed model is different from lovbjerg model since the local search algorithm is inbuilt in memetic algorithms.

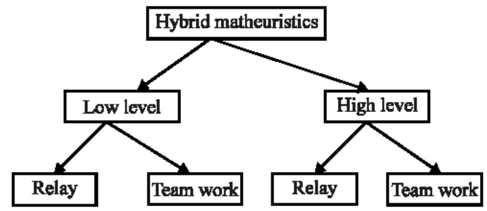

Fig. 1: Classification of hybrid metaheuristics

```
Procedure Adaptive Life Cycle Model
Begin
Initialize each PSO particle of population size 10
Initialize each MA individual of population size 0
Compute fitness value for each particle and MA individuals
Repeat
    Repeat
            For each particle
            Compute the velocity of the particle
            Move the particle to its new position
            End
            Compute fitness value for each particle
            Identify the particle with best fitness
    Until no improvement for best fitness for N iterations
    Convert particles to MA members
    Repeat
            Perform Selection on MA population
            Perform crossover on MA population
            Perform mutation MA population
            Compute fitness values for each member of MA
            population
            Identify the number with best fitness
            Until there is no change in the best fitness for N
            iterations
            Convert each member of MA population PSO particle
            Perform PSO algorithm
     Until a certain number of iterations
     Return the member of particles or MA with best  fitness as the
     rule
End
```

Fig. 2: Novel adaptive life cycle model

The fitness function: The quality of a rule set is the number of instances correctly classified (true positives).

But in the Michigan approach individual rules are evaluated independently rather than the entire rule set. This makes the false positives, number of instances classified incorrectly, also an important factor. As the individual rules of the rule set are applied in the order that are added to the rule set, a high number of false positives will negatively affect the quality of the following rules. Sousa *et al.* (2003) suggested the following fitness function which takes, not only true positives and true negatives, but also false positives and false negatives into consideration:

$$\text{Fitness} = \frac{TP}{TP+TN} * \frac{TN}{FP+TN} \tag{1}$$

Where:

TP - True Positives: Number of instances the rule classified correctly.

FP - False Positives: Number of instances the rule classified incorrectly.

TN - True Negatives: Number of instances the rule "not classified" correctly.

FN - False Negatives: Number of instances the rule "not classified" incorrectly.

Increasing the penalization of false positives has also been tested by simply increasing the weight of false positives. The fitness function (Eq. 1) explained above has been used in the heuristics of all the algorithms tested in this study.

The PSO model: The PSO model used in the Adaptive Life Cycle model is similar to the traditional PSO model described in Kennedy and Eberhart (1995). The model consists of a number of particles moving around in the search space, where the position of each particle represents a candidate solution to a numerical problem. Each particle has a position vector x_i, a velocity vector v_i and the position of the best candidate solution encountered by the particle p_i. The PSO also stores the overall best found point p_g. The memorized positions are used to attract particles to search space areas with known good solutions. In each iteration the velocity of each particle is updated in the following way (Eq. 2).

$$v_i = \chi(wv_i + \varphi_{1i}(p_i - x_i) + \varphi_{2i}(p_g - x_i)) \tag{2}$$

where, χ is known as the constriction coefficient described in Clerc (1999) and w is the inertia weight described in Shi and Eberhart (1998). Φ_1 and Φ_2 are random values, which are different for each particle and for each dimension. The velocity v_i of each particle is limited by an upper threshold v_{max}. The position of each particle is updated in each iteration by adding the velocity vector to the position vector, such that (Eq. 3),

$$x_i = x_i + v_i \tag{3}$$

The particles have no neighborhood restrictions, meaning that each particle can affect all other particles. This neighborhood is fully connected (star), which has been shown to be a good topology (Kennedy, 1999).

The MA model: A memetic algorithm consists of a population of individuals refining their candidate solutions through interaction and adaptation. Each individual represents a candidate solution to the given problem. While searching the search space, the tabu search algorithms avoids the searching from trapping in to local maxima. The MA enters a loop, in which the population is evaluated, a new population is selected and this new population is altered (Michalewicz and Fogel, 2000) after the initialization. MA used in this adaptive model uses Roulette wheel selection (Michalewicz, 1992) to generate a new population and elitism to ensure the survival of the individual with the best fitness. As in GA, the MA alters the population by crossover and mutation. The crossover operator used in the MA is the single point crossover. This operator replaces 2 parent individuals selected for crossover with 2 child individuals as follows (Eq. 4 and 5):

$$x_{child1} = w * x_{parent1} + (1-w) * x_{parent2} \tag{4}$$

$$x_{child2} = w * x_{parent2} + (1-w) * x_{parent1} \tag{5}$$

where, w is a random value between zero and one. The crossover probability PC determines the probability of an individual to be selected for crossover. For each dimension the probability of mutation PM determines whether or not to mutate. The mutation scheme used in this MA model is the non-uniform mutation described in Michalewicz and Fogel (2000).

$$\Delta_{xj} = \begin{cases} +(Max - x_j)(1 - r(1-t/T)^b) \\ -(xj - Min)(1 - r(1-t/T))^b \end{cases} \tag{6}$$

with a 50% chance each. Max is the search space maximum, Min is the minimum, r is a random number in [0, 1], t is the current iteration, T is the total number of iterations and b is a parameter determining the degree of iteration number dependency. Hence, the effect of mutation decreases over the course of the iterations with this scheme.

Experimental settings: In the experiments, we compared the performance of the standard PSO, the standard GA and the life cycle model on four benchmark datasets (Table 2). The initial population of MAs and PSOs is usually uniformly distributed over the entire search space. In all experiments the population of the novel adaptive life cycle model was fixed at 10 individuals. These are all initialized as PSO particles at the beginning.

Settings for the PSO: In the PSO model the upper limits for Φ_1 and Φ_2 were set to 2. The inertia weight ù was linearly decreased from 0.8 to 0.5 and the constriction coefficient was set to 1. The maximum velocity v_{max} of each particle was set to half the length of the search space for each dimension (v_{max} = 100). Previous research by Shi and Eberhart (1998) regarding scalability of the standard PSO showed that the performance of the standard PSO is not sensitive to the population size (Lovbjerg *et al*., 2001; Krink *et al*., 2002). In all PSO experiments, the population size is fixed as 10 particles.

Settings for the MA: For the MA, the crossover and mutation probabilities and other tabu parameters are shown in Table 1.

Data sets used in the experiments: The experiments are carried out with four public domain data sets: Breast Cancer, Wisconsin Breast Cancer, Tic-Tac-Toe and Hepatitis. These data sets were obtained from the UCI-Machine Learning Repository (Murphy and Aha, 1994). The main characteristics of each of these data sets are described in Table 2. The data sources used were also obtained from the Department of Computer Science, University of Waikato, Hamilton, New Zealand and Information and Computer Science, University of California.

Table 1: Memetic algorithms parameter settings

Parameter	Setting
Population size	10
Maximum number of generations	300
Crossover probability	0.8
Mutation probability	0.05
Length of tabu list	7

Table 2: Data sets

Data set	Cases	Categorical attributes	Continuous attributes	Classes
Ljubljana breast cancer	282	9	-	2
Wisconsin breast cancer	683	-	9	
Tic-tac-toe	958	9	-	2
Hepatitis	155	13	6	2

Table 3: Results for breast cancer data set

Algorithm	Accuracy (%)	Size of the rule set
GA	90	8
PSO	89	7
PSO+MA relay Hybrid	87	7
MA+PSO relay Hybrid	94	7

Table 4: Results for wisconsin breast cancer data set

Algorithm	Accuracy (%)	Size of the rule set
GA	90	8
PSO	89	7
PSO+MA relay Hybrid	89	7
MA+PSO relay Hybrid	92	7

Table 5: Results for Tic- Tac- Toe data set

Algorithm	Accuracy (%)	Size of the rule set
GA	90	6
PSO	89	8
PSO+MA relay Hybrid	88	6
MA+PSO relay Hybrid	89	6

Table 6: Results for Hepatitis data set

Algorithm	Accuracy (%)	Size of the rule set
GA	90	5
PSO	89	4
PSO+MA relay Hybrid	85	4
MA+PSO relay Hybrid	89	4

RESULTS

All experiments are performed for 200 evaluations. The results obtained for the four benchmark databases are tabulated in Table 3-6. The performance of the novel adaptive life cycle model compared to the standard PSO and the standard GA algorithms. Also the proposed algorithm which is based on the relay hybrid runs twice by just changing the order of the participant algorithms, i.e PSO and MA. The order of MA preceded the PSO performs better than the vice versa because of the contribution of MA by avoiding the search trapped in to local maxima. Even though both Novel Adaptive Life Cycle and MA models converged slower than PSO, proposed Novel Adaptive Life Cycle Model clearly outperformed by the standard GA and the PSO models in terms of predictive accuracy.

With miner modifications in the algorithm designed for relay hybrid, experimentation with a teamwork hybrid version could also been performed.

DISCUSSION

The proposed approach of combining two standard adaptive optimization algorithms into one self-adaptive hybrid approach turned out to be an improvement over the individual algorithms. The results showed that the novel adaptive life cycle heuristic has better performance on all benchmark datasets that we used in this study in contrast to the other adaptive algorithms, which have a highly problem dependent performance. By the observation of the experimental results it is evident that the Novel Adaptive Life Cycle Model helps the standard models to achieve the rules with high accuracy.

Based on the Talbi's taxonomy for the hybrid algorithms, combinations other than the high level relay could also be used for further research in this direction. Optimization of different parameters used in both PSO and MA would be another interesting way to pursue the further research.

REFERENCES

Angeline, P.J., 1998. Evolutionary Optimization Versus Particle Swarm Optimization: Philosophy and Performance Differences. In: Porto, V.W., N. Saravanan, D. Waagen and A.E. Eiben (Eds.). Evolutionary Programming VII. Lecture Notes in Computer Science, Springer, Berlin, 1447: 601-610.

Clerc, M., 1999. The Swarm and the Queen: Towards a Deterministic and Adaptive Particle Swarm Optimization. In: Angeline, P.J., Z. Michalewicz, M. Schoenauer, X. Yao and A. Zalzala (Eds.). Proc. Congress of Evolut. Comput. IEEE. Press, 3: 1951-1957.

Kennedy, J. and R.C. Eberhart, 1995. Particle swarm optimization. In: Proc. IEEE. Int. Conf. Neural Networks, 4: 1942-1948.

Kennedy, J., 1999. Small Worlds and Mega-Minds: Efects of Neighborhood Topology on Particle Swarm Performance. In: Angeline, P.J., Z. Michalewicz, M. Schoenauer, X. Yao, A. Zalzala, (Eds.). Proc. Congress Evolut. Comput. IEEE Press, 3: 1931-1938.

Krink, T. and M. Lovbjerg, The Life Cycle Model: Combining Particle Swarm Optimization, genetic Algorithms and HillClimbers. In: Proc. parallel Solving form Nature, VII (PPSN-2002): 621-630.

Krink, T., R. Thomsen and P. Rickers, 2000. Applying Self-Organised Criticality to Evolutionary Algorithms. In: Schoenauer, M., K. Deb, G. Rudolph, X. Yao, E. Lutton, J.J. Merelo, H.P. Schwefel, (Eds.). Parallel Problem Solving from Nature-PPSN VI. Springer, Berlin, pp: 375-384.

Krink, T., J.S. Vesterstrfm and J. Riget, 2002. Particle Swarm Optimisation with Spatial Particle Extension. In: Fogel, D.B., X. Yao, G. Greenwood, H. Iba, P. Marrow and M. Shackleton (Eds.). Proces 4th Congress on Evolutionary Computation (CEC).

Lawrence, E., 1996. Henderson's Dictionary of Biological Terms. Longman.

Lovbjerg, M., T.K. Rasmussen and T. Krink, 2001. Hybrid Particle Swarm Optimiser with Breeding and Subpopulations. In: Spector, L., E.D. Goodman, A. Wu, W. Langdon, H.M. Voigt, M. Gen, S. Sen, M. Dorigo, S. Pezeshk, M.H. Garzon, E. Burke (Eds.). Proc. Genetic and Evolut. Comput. Conf. (GECCO) Morgan Kaufmann, pp: 469-476.

Michalewicz, Z., 1992. Genetic Algorithms + Data Structures = Evolution Programs. Springer, Berlin.

Michalewicz, Z. and D.B. Fogel, 2000. How to Solve It: Modern Heuristics. Springer, Berlin.

Murphy, P.M. and D.W. Aha, 1994. UCI Repository of Machine Learning Databases. http://www.ics.uci.edu/~ mlearn/MLRepository.html. Irvine, CA: University of California, Department of Information and Computer Science.

Parpinelli, S.R., H.S. Lopes and A.A. Freitas, 2002. An Ant Colony Algorithm for Classification Rule Discovery in Data Mining: A Heuristic Approach. Londan: Idea Group Publishing, pp: 191-208.

Shi, Y. and R.C. Eberhart, 1998. Parameter Selection in Particle Swarm Optimization. In: Porto, V.W., N. Saravanan, D. Waagen, A.E. Eiben (Eds.). Evolutionary Programming VII. Lecture Notes in Computer Science 1447. Springer, Berlin, pp: 591-600.

Sousa, T., A. Neves and A. Silva, 2003. Swarm Optimization as a New Tool for Data Mining. Proceedings of International Parallel and Distributed Processing Symposium.

Suganthan, P., 1999. Particle Swarm Optimiser with Neighbourhood Operator. In: Angeline, P.J., Z. Michalewicz, M. Schoenauer, X. Yao, A. Zalzala, (Eds.). Proc. Congress of Evolut. Comput. IEEE., 3: 1958-1962.

Talbi, E.G., 2002. Ataxonomy of hybrid metaheuristics. Kluwer Academic Publishers. J. Heuristics, 8: 541-564.