Eurygaster Algorithm: A New Approach to Optimization

Fariborz Ahmadi
Department of computer science,Ghorveh Islamic azad university, Iran,Ghorveh branch

Hamid Salehi
Department of computer science,Ghorveh Islamic azad university, Iran,Ghorveh branch

Khosro Karimi
Department of computer science,Ghorveh Islamic azad university, Iran,Ghorveh branch

ABSTRACT
Almost all of the approach to solve NP-hard and NP-complete problem simulate artificial life. In this research, the behavior of eurygaster life is studied, so according to their life the new algorithm is introduced. In spite of PSO algorithm, that is used to solve continuous nonlinear functions, researchers' algorithm is so suitable to solve both continuous and discrete functions. Eurygasters attack to grain farms and distributed over them. It is worth to mention that these insects attack to farms in groups furthermore each group colonizes in one farm. It is observed that after periods of time all of the farms in a region are occupied by these groups of eurygasters. When each group of these insects are going to seek a farm to feed on it, they consider nearly all the farms and settles on a farm which have a lowest distance with them and doesn't have any group of eurygasters. It is clear that by distributing several groups of eurygasters, depending on the problem size, on search space of problem, the solution of the problem can be extracted. In this research, using the behavior of eurygasters, a new algorithm has been invented and has been tested on graph partitioning. The evaluation results show the advantage of researcher algorithm over ancient ones like genetic and PSO.

General Terms
Evolutionary Computation

Keywords
Evolutionary Computation, genetic algorithm, particle swarm optimization, eurygaster algorithm, evolutionary programming.

1. INTRODUCTION
Nonlinear problems are important in many real life applications because of their intrinsic difficulty; recently the area has attracted much research with the advances in nature inspired heuristics and multi agent systems for these problems. The dramatic increase in the size of the search space and the need for real time solutions motivated research idea in solving NP-problems using nature inspired heuristic techniques.in order to solve NP-problems some method like genetic algorithm and particle swarm optimization has been invented. These algorithms are population based and use intelligent to converge to the solution of problems.

Genetic algorithms consist of a certain number of individuals in each generation. The individuals are called chromosomes and each one show one of the solutions of problem. In each generation the fitness of chromosomes are evaluated and crossover and mutation operators are applied to the selected chromosomes to generate new chromosomes or individuals (offspring) in the next generation.

Also particle swarm optimization [1][2] was introduced by Kennedy and Eberthart. In PSO, Some social systems of natural species, such as bird flock and fish School, possess interesting collective behavior. In these systems, globally sophisticated behavior emerges from local, indirect communication among simple agents with only limited capabilities. In an attempt to simulate this flocking behavior by computers, Kennedy and Eberthart realized that an optimization problem can be formulated as that of a flock of bird's flying across an area seeking a location with abundant food. This observation, together with some abstraction and modification techniques, led to the development of a novel optimization technique – particle swarm optimization.

To solve NP-problem some other algorithm such as ant colony [3], simulated annealing [4], tabu search [5], honey bee algorithm [6], photosynthetic algorithm [7], enzyme algorithm [8], glowworm swarm optimization [9], monkey search [10], firefly algorithm[11] were also introduced. These algorithms simulate behavior of artificial life of species.

In this work, researchers have invented new method called eurygaster algorithm based on eurygaster life. This algorithm is suitable for both continuous and discrete NP-problems and also, special for divide and conquer problems. In researcher approach, to find the solution of problem, problem must be divided in to some categories. After that, each cattle of eurygasters scatters over one of these categories to find the solution in search space of the category. In case of not finding the solution, new eurygasters are produced and distributed on the other categories and in case of finding the solution, the algorithm is finished. This method is easy to implement and can be simulated by a few line of computer code and is computationally inexpensive in term of memory and speed. Despite PSO that is used for continuous NP-problem, this algorithm is proposed for both continuous and discrete NP-problems. In other hand, in PSO algorithm [1][2], particles go through the search space of problem at each time continuously, while in research approach a group of eurygasters change their presentations and locations according to new category in which they colonize. In researcher approach, the search space of problems must be split on several groups or categories so that a great number of eurygasters disperse on each group to exhaustedly consider its space to find the best solution in that group.

The reminder of this paper is divided into 5 sections. At first in section 2, the characteristics of previous work like PSO and GA are detailed. After that, in section 3 behavior of eurygaster is described. Then, in section 4, our method or eurygaster algorithm is elaborated. Finally, section 5, 6 show the evolution results of our method on graph partitioning [12] and draw some conclusion, respectively.

2. Previous works

PSO and GA have been interested by researchers in the area of heuristic and soft computing methods. In this paper the characteristics of both of them are described and finally, we compare our approach with these methods to show the efficiency of our approach over them.

2.1 Overview of PSO

Particle swarm optimization (PSO) is an algorithm modeled on swarm intelligence, that finds a solution to an optimization problem in a search space, or model and predict social behavior in the presence of objectives. The PSO is a stochastic, population-based computer algorithm modeled on swarm intelligence. Swarm intelligence is based on social-psychological principles and provides insights into social behavior, as well as contributing to engineering applications. The particle swarm optimization algorithm was first described in 1995 by James Kennedy and Russell C. Eberhart [1][2]. By adding a new inertia weight into PSO, a new version of PSO is introduced in [13].

The particle swarm simulates a kind of social optimization. A problem is given, and some way to evaluate a proposed solution to it exists in the form of a fitness function. A communication structure or social network is also defined, assigning neighbors for each individual to interact with. Then a population of individuals defined as random guesses at the problem solutions is initialized. These individuals are candidate solutions. They are also known as the particles, hence the name particle swarm. An iterative process to improve these candidate solutions is set in motion. The particles iteratively evaluate the fitness of the candidate solutions and remember the location where they had their best success. The individual's best solution is called the particle best or the local best. Each particle makes this information available to their neighbors [14].

They are also able to see where their neighbors have had success. Movements through the search space are guided by these successes, with the population usually converging, by the end of a trial, on a problem solution better than that of non-swarm approach using the same methods. Each particle represents a candidate solution to the optimization problem. The position of a particle is influenced by the best position visited by itself i.e. its own experience and the position of the best particle in its neighborhood i.e. the experience of neighboring particles. When the neighborhood of a particle is the entire swarm, the best position in the neighborhood is referred to as the global best particle, and the resulting algorithm is referred to as the gbest PSO. When smaller neighborhoods are used, the algorithm is generally referred to as the pbest PSO. The performance of each particle is measured using a fitness function that varies depending on the optimization problem. Each Particle in the swarm is represented by the following characteristics:

- The current position of the particle
- The current velocity of the particle

The particle swarm optimization which is one of the latest evolutionary optimization techniques conducts searches uses a population of particles. Each particle corresponds to individual in evolutionary algorithm. Each particle has an updating position vector and updating velocity vector by moving through the problem space.

2.2 Overview of genetic algorithm

GA is inspired from the natural selection of Darwin to find the solution of the problems. The simplest form of genetic algorithm involves three types of operators: selection, crossover (single point), and mutation [16].

Selection: This operator selects chromosomes in the population for reproduction. The fitter the chromosome, the more times it is likely to be selected to reproduce [17].

Crossover: This operator randomly chooses a locus and exchanges the subsequences before and after that locus between two chromosomes to create two offspring. The crossover operator roughly mimics biological recombination between two single−chromosome (haploid) organisms [18].

Mutation: This operator randomly flips some of the bits in a chromosome. Mutation can occur at each bit position in a string with some probability, usually very small [17].

The simplest form of genetic algorithm involves three types of operators: selection, crossover (single point), and mutation [16].

3. Eurygaster behaviors

Eurygaster integriceps is an insect pest that predominantly attacks grains, feeding on the leaves, stems and grains, reducing yield and injecting a toxin into the grains which adds a foul smell to the resulting flour, and substantially reduces the baking quality of the dough.

In winters eurygasters live under the plants and bushes in hillside, in several numbers and make a group. At the end of winter and at the beginning of spring when it gets warmer, these insects end their winter sleeps and get ready to move and fly to grain fields by moving over the high mountains and leaving the nests in groups. The first group by the use of its instinct finds the best and the nearest grain fields and stays there. Getting there, this group of insect sends signals to the air to show the other groups their being there. Based on the number of eurygasters in a place, the strength of signals will be different. If the number of eurygasters in a grain field is not great, the rate of diffused signals will be little and if the number of eurygasters in a grain fields is greater, the rate of diffused signals will be increased. They diffuse these signals to show the others that reside there. So that the other groups of eurygasters understand that they should not close to the grain field which contains the first group. Of course the other groups based on diffused signals by the first group and the strength of these signals they decide if they can land and stay there or not. If the power of diffused signals is low, it means that some of the other groups of eurygasters can land and stay by the other groups which are resident there and began to eat. While the strength of the signals in the sky is high, it means that the other groups cannot land on the field(s) containing eurygasters, and they must fly to other fields in which there are no eurygasters, to live and eat.

According to the passage mentioned above, the next group of eurygasters while flying from their nests to other fields to find the best grain fields searches the best and closest ones to land and eat based on the broadcasted signals by landed group(s). This process will continue until they will find a suitable and useful grain field to eat.

We conclude that all the grain fields in a wide area will be attacked by eurygasters, because they do not gather in a one place.so, when there is not enough food in a grain field in which the eurygasters have stayed for a time, they will fly to a

new field with no eurygasters according to the process mentioned above.

4. Eurygaster algorithms

In this section, proposed approach or eurygaster algorithm is described. Solving non-linear functions are so necessary in real life today and recently researchers interested in inventing methods to solve them. Thus, our approach contributes to solve NP-class problems. The great advantage of this algorithm is that it's so easy to implement and is also inexpensive in term of memory and speed. The second advantage of this algorithm is its convergence speed compared to other methods like GA and PSO.

In spite of GA that traps in local optimum, the evaluation results of this approach show this method doesn't have this drawback. The major disadvantage of this method is that it can be used only for dividable problems. So the problems must be divided into some groups to be solved by researchers approach. Despite this problem, the algorithm advantage clearly outweighs over its disadvantage.

As it was mentioned, eurygasters attack wheat fields in groups. When the new generation is produced, they attack the fields which haven't been attacked before. After some period of times, it is revealed that all the fields in a region have been attacked totally by eurygasters. The main point inspired by eurygasters behaviors is that if we can divide the problems into some partitions by producing element named eurygasters in each partition all the space of the problem can be searched. The main point in solving problems by this algorithm is the way in which the problem can be divided into some partitions or how to the problem can be partitioned. That is, the better the partitioning is done, the more accurate the solutions to the problem can be concluded.

Now suppose that the problem has been divided into n partitions. First several eurygasters based on the size of the problems partitions are generated and distributed in the space of the first part to find the solution of the problem. It should be mentioned that if the number of eurygasters is not enough to exhaustedly cover all the space of the related partition, we can search all the related space by redistributing them and changing their position so that they can cover all the space mentioned. This process in genetic algorithms is done through mutation [15]. Using mutation in the proposed approach prevents local optimum. After searching for the first partition and in case of not finding the optimal solution to the problem, new kinds of eurygaster relevant to the second part are produced in order to be used to search in the space of the second partition. This operation is continued up to the n-part and in case of finding the expected solution in each part, the related algorithm is terminated and the solution of the problem is reported. The related semi-code of the proposed algorithm is as follows:

I← the number of clusters

While I <> 0 do

1. Initialization: produce euragasters or particles according to characteristic of one partition
2. Distribution: distribute eurygasters on the regions of the partition
3. Evaluation: evaluate suitability of each eurygaster or particle depend on the problem
   - 3.1 If the suitable result of the partition is not obtained
     - 3.1.1. Change the position of Eurygasters in the partition
     - 3.1.2. goto 3
   - 3.2 If the result of the problem is not obtained
     - 3.2.1. I--
     - 3.2.2. goto 1

     Else
     - 3.2.3. Stop algorithm or break

   End while
4. Report the solution of the problem

Algorithm. 1 .Eurygaster algorithm

In proposed algorithm by researchers, namely eurygaster algorithm, the partitions must be created before execution of the algorithm. At first, in *initialization* phase a set of the eurygasters are produced and scattered on the search space of relevant partition by *distribution* phase. After that the suitability of each eurygaster is computed according to the problem kind as you can see in evaluation phase. In case of finding the solution of the problem in any of the partition, there may be two possibilities.

a. The found solution is optimal solution for that partition but not for the main problem. In this case line 3.1 is not executed but 3.2 is executed and the algorithm executes initialization phase to create new eurygasters for searching the solution in another part because of moving control to that phase in line 3.2.2 (goto 1).

b. The found solution is optimal solution for that partition and also for the main problem. In this case neither the sub instructions of line 3.1 nor the sub instructions of line 3.2 are executed and cause the algorithm be stopped. Therefore, this is a point that the absolute solution of the problem is reached.

In case of not finding the best solution of the partition or when it is hoped that the solution is in search space of the partition, line 3.1 is executed and changed the eurygasters location to fully cover the area of the pertition search space. Finally, if the appropriated solution to the problem is obtained, the algorithm is terminated by line 3.2.3, otherwise a new partition is created and also new eurygasters are produced to be distributed over the partition.

5. Evaluation Results

The results of the studies show the algorithm PSO is suitable for continuous problems and its using in discrete problems is very rare and expensive, while the proposed algorithm is suitable for both continuous and discrete problems. In the proposed algorithm the space of the problem is divided into several partitions and in each step of algorithm just one partition of the problem is investigated and in case of not reaching the expected solution, the other part is going to be searched. Also, in each part by changing the position of the eurygasters, we can prevent trapping in local optimum. On the one hand, our method has great advantages over genetic algorithms. In the genetic algorithms, the chromosomes of the first generation are distributed randomly all over the problem, this operation causes some problems as follows:

1. The probability of trapping in the local optimum increases.
2. Chromosomes that will be produced in the next generations can be placed in some locations in the problem space where the chromosomes of the previous generations had been placed. In other word, some regions of the problem space will be searched several times that

causes execution time of the algorithm is increase and lead to decreasing in convergence speed.

3. Since in each generation all the space of the problems is searched for, the execution time of the algorithm will be increased. The proposed algorithm will not have the above-mentioned problems because of searching all the partition cotinuously, orderly and separately.

To evaluate the proposed algorithm we have used "Graph partitioning" [12] in which the nodes are split into some clusters in a way that the amount of interconnection is decreases to the least and intra connection increases to the most. To demonstrate the optimum of the proposed method, 100 nodes have been used to partitions into 3, 4, 5 and 6 clusters.

Before solving the problem by using the proposed algorithm, the related partitioning to solve the problems or the wheat fields must be determined. For example to solve the problem of "Graph Partitioning" to three clusters the following partitions can be done. Suppose n1, n2, and n3 are the number of the related nodes of clusters 1, 2 and 3 respectively.

n1>n2>n3 , n1>n2= n3, n1=n2>n3 , n1=n2=n3

Similarly, this kind of division can be done for the above-mentioned problem into 4, 5 and 6 clusters. The following figure1 shows the execution time of the above-mentioned problem in case of dividing into 3, 4, 5, and 6 clusters by using the genetic algorithm and the proposed algorithm.

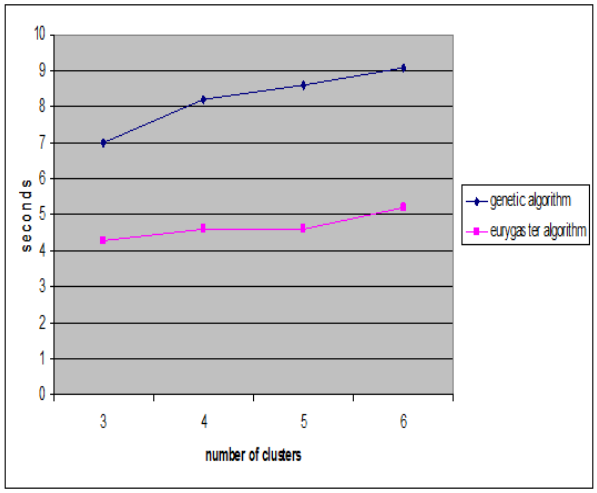

Fig. 1. Execution time of the graph partitioning problem

In Fig. 1, the axis of X shows the number of the clusters and the axis of Y shows the execution time of the algorithm in seconds. As the above Fig. 1 shows the execution time of the proposed algorithm is much less than the execution time of the genetic algorithm. Moreover, every program has been operated 100 times which Fig. 2 shows the percent of getting the exact solution to the problem in both of the algorithms.

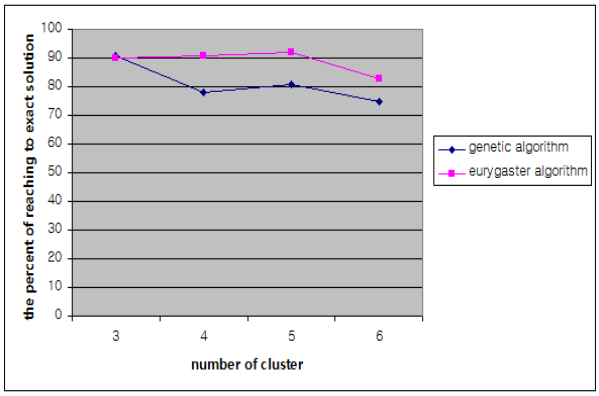

Fig. 2. Percent of getting the exact answer to the problem

Since the limitation that local optimum has created for the genetic algorithm, the percentage of successfulness by using this algorithm to get the exact solution is less than the proposed algorithm. Proposed algorithm lacks the limitations of local optimum and if the related fitness functions are defined properly, it can be claimed that almost 100 percent of cases getting the exact solution is possible. Also, Fig. 3 show that our approach has totally more performance than genetic algorithm. This figure has been computed by multiplying of Fig. 1 and Fig. 2 and denotes the ratio of the algorithm in reaching the exact solution of problems. As you can see in the Fig. 3, the total time of our approach is less than genetic algorithm that can be concluded researcher approach is faster and convergence speed is high

.

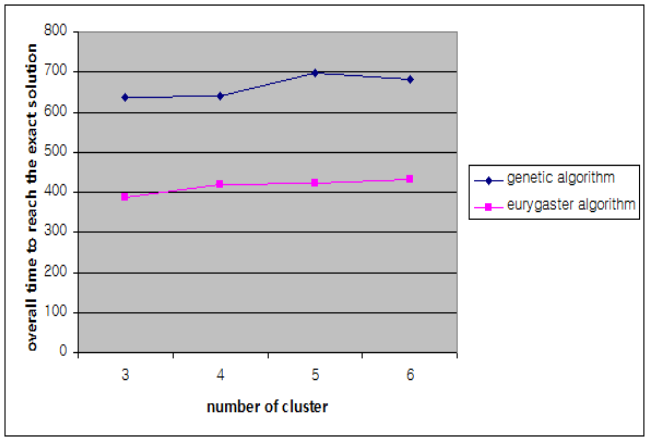

Fig. 3. Overall time to reach the exact solution

6. Conclusions

In this article, a new method based on the behaviors of eurygasters has been presented to solve NP-class problems. Although PSO algorithms just can be used for continuous problems, this method can be used for both continuous and discrete problems. This approach unlike genetic algorithm lacks the local optimum so the probability of getting the more accurate solution in this method is muchmore than the genetic algorithm. Moreover, in this algorithm every space of the problem is searched for once while in the genetic algorithm every part of the problem space can be searched several times in different generations, so the rate of convergence in this algorithm is much more than the genetic algorithm. This algorithm is a suitable replacement for the genetic algorithm.

Also, this algorithm is easy to implement by computer. If takes a few lines to programming and doesn't need a huge memory or CPU speed. In the proposed algorithm the space of the problem is divided to several partitions and every partition is searched for separately. The more efficient manner to reach the more convergence speed is that if the solution of a problem is found in a specific partition, the searching process be stopped in order to decrease the amount of execution time of the algorithm. This algorithm can be used for Divide and Congers problems properly. Since different parts of the problem have been separated, this method is very simple and efficient to be applied in parallel systems. The researchers hope that the explanation in this article can satisfy its readers about the operation of the algorithm.

7. REFERENCES

[1] R.C. Eberhart, and J. Kennedy, "A new optimizer using particle swarm theory". In Proceedings of the sixth international symposium on micro machine and human science, volume 43. New York, NY, USA: IEEE, 1995.

[2] J. Kennedy, and R.C. Eberhart, "Particle swarm optimization". In Proceedings of IEEE international conference on neural networks, volume 4, pages 1942–1948. Perth, Australia, 1995.

[3] M. Dorigo, "Optimization, Learning and Natural Algorithms", PhD thesis, Politecnico di Milano, Italie, 1992.

[4] S. Kirkpatrick, C. D. JR. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing", IBM Research Report RC 9355.

[5] F. Glover, and M. Laguna, "Tabu Search", Kluwer Academic Publishers, Boston, 1997.

[6] Pham, DT. Ghanbarzadeh, A. Koc, E. Otri, S. Rahim, and M. Zaidi, "The Bees Algorithm. Technical Note, Manufacturing Engineering Centre, Cardiff University, UK, 2005.

[7] H. Murase, "Finite element analysis using a photosynthetic algorithm", Computers and Electronics in Agriculture, 29, pp. 115-123, 2000.

[8] X. S. Yang, " New enzyme algorithm", Tikhonov regulation and inverse parabolic analysis, in Advances in Computational Methods in Science and Engineering, Lecture Series on Computer and Computer Sciences, ICCMSE, Eds. T. Simons and G. Maroulis, 4, 1880-1883, 2005.

[9] K.N. Krishnanand, and D.Ghose, "Detection of multiple source locations using a glowworm metaphor with applications to collective robotics," IEEE Swarm Intelligence Symposium, Pasadena, California, USA, pp. 84–91, 2005.

[10] A. Mucherino, and O. Seref, "Monkey Search: A Novel Meta-Heuristic Search for Global Optimization,"AIP Conference Proceedings 953, Data Mining, System Analysis and Optimization in Biomedicine, 162–173, 2007.

[11] X.S. Yang, "Firefly algorithm," (chapter8) in: Nature-inspired Metaheuristic Algorithms, Luniver Press, 2008.

[12] D. Doval, S. Mancoridis, and B. Mitchell,"Automatic clustering of software systems using a genetic algorithm," STEP '99, IEEE Computer Society, 1999.

[13] Y. Shi, and R. Eberhart, "A Modified Particle Swarm Optimizer," IEEE International Conference on Evolutionary Computation, Anchorage, Alaska, May 4-9, 1998.

[14] P. Mathiyalagan, U.R. Dhepthie, and S.N. Sivanandam, "Grid Scheduling Using Enhanced PSO Algorithm," (IJCSE) International Journal on Computer Science and Engineering, Vol. 02, No. 02, 140-145, 2010.

[15] R. J. Collins, and D. R. Jefferson, "The evolution of sexual selection and female choice," In F. J. Varela and P. Bourgine, eds., Toward a Practice of Autonomous Systems: Proceedings of the First European Conference on Artificial Life. MIT Press, 1992.

[16] D. Ackley, and M. Littman, "Interactions between learning and evolution," In C. G. Langton, C. Taylor, J. D. Farmer, and S. Rasmussen, eds., Artificial Life II. Addison−Wesley, 1992.

[17] L. Altenberg, "The evolution of evolvability in genetic programming," In K. E. Kinnear, Jr., ed.,Advances in Genetic Programming. MIT Press, 1994.

[18] R. M. French, and A. Messinger, "Genes, phenes, and the Baldwin effect: Learning and evolution in a simulated population," In R. A. Brooks P. Maes, eds., Artificial Life IV. MIT Press, 1994.