SWARM-BASED OPTIMISATION

LENKA RAUDENSKÁ

1 INTRODUCTION

Swarm intelligence is based on nature-inspired behaviour and is successfully applied to optimisation problems in a variety of fields. The goal of optimisation is to find the optimum in the smallest possible amount of iterations, where optimum means the best from all possibilities chosen from a particular point of view (so-called: criterion).

2 SWARM-BASED INTELLIGENCE

Swarm-based intelligence is artificial intelligence technique based on the study of collective behaviour in self-organizing systems.

Swarm-based systems are usually composed from population of individual, which takes effect between each other and environment. Individual could communicate directly or through impacting in surroundings [1].

Although this systems do not have any central control of the individual behaviour, interaction between individuals and simple behaviour between them usually lead to detection of aggregate behaviour, which is typical for whole colony.

This could by observe by ants, bees, birds or bacteria in the nature. By inspiration of these colonies were develop algorithms called Swarm-based intelligence and are successfully apply for solving complicate optimisation problems [3].

2.1 Genetic algorithm (GA)Preparation for publication

Genetic algorithm is search based on the natural evolutionary process and is a stochastic search technique which works on the process of natural selection.

GA begins with an initial population (i.e. the first generation), which is usually generated randomly. This population evolves the next generation, which expectantly contains better solutions (fitness) [10].

There are use three fundamental operators: selection, recombination (also called crossover) and mutation.

ISSN 1335-1745

Value of fitness determines how good the candidate is. The selection operator presents natural selection, it means that bigger chance to be selected is for candidate with higher fitness. The best selected candidates create the next generation using the recombination operator. The recombination is applied on pairs (called parents) and is producing two children solutions (for each pair). Each new child contains a part of both parents. Practically it is usually arithmetical mean.

The mutation operator randomly chose a few candidates from the generation and changes its value in chromosome and product is children solutions. In practice it is based on adding random vector to chromosome.

Both of the operators (recombination and mutation) should by conform to make solution of new generation even better compared to old one. New generated population of children become for next generation parent [3].

The basic steps of a GA:

1) Random generate the initial population.
2) Evaluate the fitness for all candidates.
3) Repeating of the following steps until a satisfactory solution is reached:
    a) Select parents from the population according to fitness value.
    b) Create children from selected parents using a recombination operator.
    c) Mutate operator use on candidate with a smaller fitness value.
    d) Create the next generation with the children.
    e) Evaluate the fitness of all candidates in the new population.

Termination of the algorithm is reaching the satisfactory solution or after given number of repetitions.

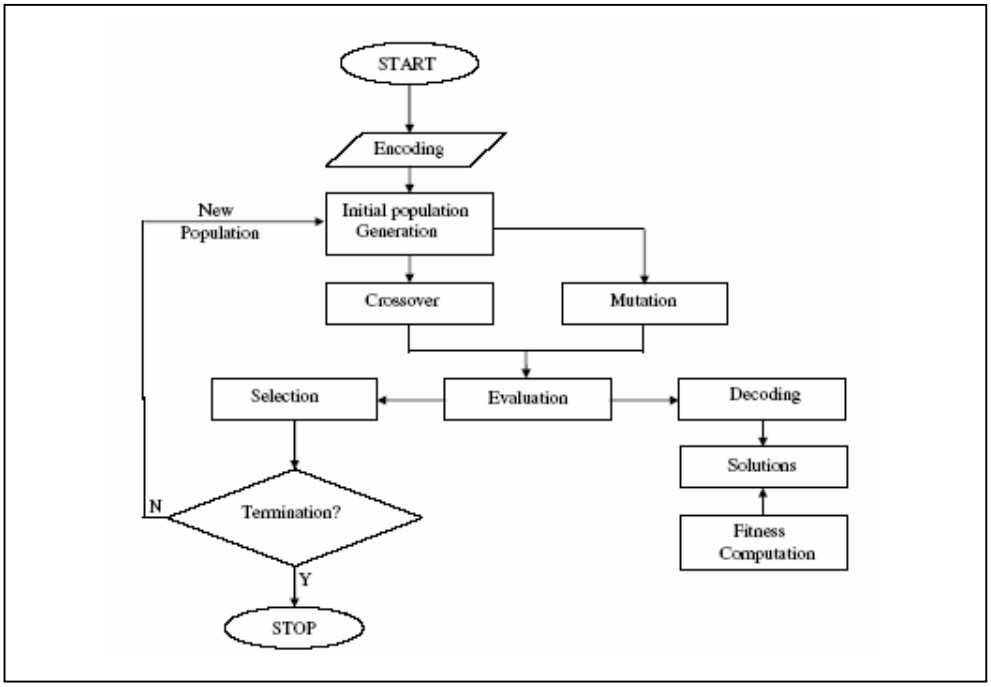

Figure 1 - Flow-process diagram of a simple genetic algorithm.

2.2 Ant Colony Optimization (ACO)

The basic principle is based on finding the shortest path from food source to anthill by smelling pheromones (chemical substances they leave on the ground during walk).

System of obtaining food in ant colony is managed by hundreds of individuals and covered thousands of square meters. In process of collecting food if there are two possible paths to reach a food source, as shown in Fig. 1, and they have no clue about which direction to choose, they choose it randomly. It is assumed that half of them choose the first direction and the rest choose the other one. Suggesting that all ants have same walking speed, the shorter way will receive a greater amount of pheromone per time. Next time when they will choose the shortest way by smelling more pheromone on the shorter path than the longer one. Other ants make use of pheromone concentration to determination of the shortest way, which give them the possibility to collect food quicker [8].

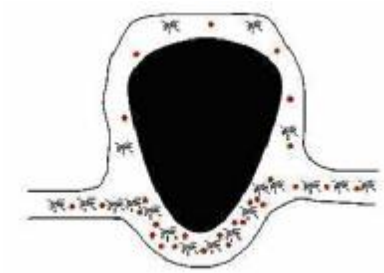

Figure 2 - The pheromone deposition of ants (red dots).

2.3 Particle swarm optimization (PSO)

Particle swarm optimization is one of the latest evolutionary optimization techniques with a stochastic population based optimization approach.

PSO is inspirited by interaction and communication in a flock of birds or shoal of fishes. In these groups, there is a leader (individual with the best value of fitness) who guides the movement of the whole swarm. The movement an individual is based on the leader and on its own knowledge. Generally could by said that the model PSO presuppose that the behaviour of each individual is a compromise between its own and collective knowledge.

The basic steps of a PSO:

1) Setting of population block data with random value of a position (i.e. a solution) and a velocity (i.e. change pattern of the solution).
2) Every individual knows its position and the value of the objective function for that position. It also remembers its own best previous position and its corresponding objective function value.
3) Evaluate the fitness of all individuals.
4) Comparing the current fitness of each individual with its own historical best position, and if its own historical best position is smaller then it is replaced with the current fitness.
5) Comparing the best current position of all individuals with the historical best position of the whole swarm, if the historical best position of the whole swarm is smaller then it is replaced with the best current position of all individuals.
6) Refreshing the positions and velocities of all individuals according to the following equations:

$$v_{i,t+1} = c_1 v_{i,t} + c_2 (p_{i,t} - x_{i,t}) + c_3 (p_{\psi i,t} + x_{i,t})$$

$$x_{i,t+1} = x_{i,t} + v_{i,t+1}$$

Variables:

xi,t ... position of individual in iteration (equivalent to the one problem solution)

vi,t ... velocity of individual in iteration (equivalent to the change pattern of the solution)

pi,t ... the best previous position among all the individuals in iteration (memorized by each individual)

7) Termination of the algorithm is reaching the satisfactory solution or after given number of repetitions.

2.4 The Bees Algorithm

Bees Algorithm is inspired by the natural behaviour of honey bees during collecting pollen.

The process begins by sending scout to search round for promising sites. Scout bees move randomly during the searching. When they return to the hive, they express by dancing three pieces of information about found site: the direction in which it is situated, its distance from the hive and its quality. This information helps the colony to evaluate the amount of energy needed to harvest it and after it they can send its bees to the most promising place directly.

Each individual's knowledge of the outside environment comes only from this dance. After dancing more follower bees are sent to more promising patches. This allows the colony to harvest pollen quickly and efficiently.

1) Random generate the initial population.
2) Evaluate the fitness for all candidates.
3) Repeating of the following steps until a satisfactory solution is reached:
   a) Select sites for circumambience search.
   b) Send out bees to selected sites (more bees for better sites) and evaluate its fitness.
   c) Select bees with the highest fitness from each patch.
   d) Assign remaining bees to search randomly and evaluate their fitness.

On the end of each iteration has swarm two parts of population – individuals from selected sides and scout bees designate for searching.

Termination of the algorithm is reaching the satisfactory solution or after given number of repetitions.

2.5 Comparison of Swarm-based optimisation algorithms

Optimization algorithms above were compared by eight following benchmark functions with hundred independent measurements.

Table 1: Methods comparison

Function	GA		ACO		Bees Algorithm	
	Number of iterations	Success [%]	Number of iterations	Success [%]	Number of iterations	Sucess [%]
De Jong	10160	100	6000	100	49	100
Goldstein & Price	5662	100	5330	100	999	100
Branin	7325	100	1936	100	1657	100
Martin & Gaddy	2844	100	1688	100	526	100
Rosenblock	10212	100	6842	100	898	100
Hyper sphere	15468	100	22050	100	7113	100
Griewangk	200000	100	50000	100	1847	100

The first function De Jong's figured out, that the Bees Algorithm reached the optimum 207 faster than GA and 120 times faster than ACO, with a success of 100%. For the next function Goldstein & Price, the Bees Algorithm could find the optimum almost 5 times faster than GA and ACO, again with 100% success. With Branin's function, there was for Bees Algorithm a 15% improvement compared with ACO and 77% improvement compared with GA, also with 100% success. Rosenbrock's function in two-dimensions has with the Bees Algorithm at least twice fewer evaluations than the other methods also with 100% success. Four-dimensions Rosenbrock's function, where ACO could reach the optimum 3,5 times faster than the Bees Algorithm with success rate 100%. In Hyper Sphere model of six dimensions, the Bees Algorithm needed half of function evaluations compared with GA and one third compared with ACO. Last but not least Griewangk function is ten-dimensional and the Bees Algorithm found the optimum with 100% success and 10 times faster than GA and 25 times faster than ACO [7].

3 CONCLUSION

This paper has presented methods of Swarm-based intelligence. From comparison above follows that the Bees Algorithm gives the results in the biggest number of function in the shortest time with 100% success rate in all cases. The Bees Algorithm is the youngest from this and is still in its beginning – it is still developing. Future work for this algorithm should turn to the reduction of parameters and incorporation of better learning mechanisms or combination with some other earlier mentioned algorithm.

REFERENCES

[1] C. Andrés, S. Lozanob. A particle swarm optimization algorithm for part–machine grouping. 2005. Robotics and Computer-Integrated Manufacturing 22 (2006) 468–474.

[2] Sheng-Fa Yuana, Fu-Lei Chua. Fault diagnostics based on particle swarm optimisation and support vector machines. 2006. Mechanical Systems and Signal Processing.

[3] Qie He, Ling Wang, Bo Liu. Parameter estimation for chaotic systems by particle swarm optimization. 2006. Chaos, Solitons and Fractals (2006).

[4] A. Kumar, Prakash, M.K. Tiwari, R. Shankar, A. Baveja. Solving machine-loading problem of a flexible manufacturing system with constraint-based genetic algorithm. 2004. European Journal of Operational Research 175 (2006) 1043–1069.

[5] Erick Cantú-Paz, D. E. Goldberg. Effcient parallel genetic algorithms: theory and practice. 1999. Comput. Methods Appl. Mech. Engrg. 186 (2000) 221-238.

[6] A. Bahreininejad, P. Hesamfar. Subdomain generation using emergent ant colony optimization. 2005. Computers and Structures 84 (2006) 1719–1728.

[7.] D. T. Pham, A. Ghanbarzadeh, E. Koc, S. Otri, S. Rahim, M. Zaidi. The Bees Algorithm – A Novel Tool for Complex Optimisation Problems. Manufacturing Engineering Centre, Cardiff University.

[8] K. Eshghi, M. Kazemi. Ant colony algorithm for the shortest loop design problem. 2006. Computers & Industrial Engineering 50 (2006) 358–366.

[9] M. Izadifar, M. Zolghadri Jahromi. Application of genetic algorithm for optimization of vegetable oil hydrogenation process. 2004. Journal of Food Engineering 78 (2007) 1–8.

[10] H. Stern, Y. Chassidim, M. Zofi. Multiagent visual area coverage using a new genetic algorithm selection scheme. 2003. European Journal of Operational Research 175 (2006) 1890–1907.

ABOUT THE AUTHOR

Ing. Lenka Raudenská, Faculty of Mechanical Engineering, Brno University of Technology, Technická 2896/2, 616 69 Brno, Czech Republic, tel.: 54114 2521, yraude00@stud.fme.vutbr.cz.