Int j simul model 6 (2007) 3, 154-164
ISSN 1726-4529 Original scientific paper

HONEY-BEES OPTIMIZATION ALGORITHM APPLIED TO PATH PLANNING PROBLEM

Curkovic, P. & Jerbic, B.
Faculty of Mechanical Engineering and Naval Architecture, Ivana Lučića 5, 10000 Zagreb, Croatia
E-Mail: petar.curkovic@fsb.hr; bojan.jerbic@fsb.hr

Abstract
Autonomous systems assume intelligent behaviour with capabilities of dealing in complex and changing environments. Problem of path planning, which can be observed as an optimization problem, seems to be of high importance for arising of intelligent behaviour for different real-world problem domains. Swarm intelligence has gained increasingly high interest among the researchers from different areas, like, science, commerce and engineering over the last few years. It is particularly suitable to apply methods inspired by swarm intelligence to various optimization problems, especially if the space to be explored is large and complex. This article presents application of Honey-bees mating algorithm (HBO) to a non linear Diophantine equation benchmark problem and comparison with results of a genetic algorithm (GA) designed for the same purpose. In second part of the work, HBO algorithm is applied to solve a problem of guidance of mobile robot through the space with differently shaped and distributed obstacles. Fuzzy fitness function for selective evaluation of paths found by the algorithm is proposed. The performance of the algorithm is comparable to genetic algorithm developed for the same purpose.
(Received in November 2006, accepted in March 2007. This paper was with the authors 1 month for 1 revision.)

Key Words: Swarm Intelligence, Mobile Robot, Optimization, Path Planning, Obstacles

1. INTRODUCTION

One of the ultimate goals in Robotics is to create *autonomous robots*. Such robots will accept high-level descriptions and tasks and will execute them without further human intervention [1]. This work presents approach where user specifies to the robot solely *what* is wanted for robot to do (find collision free path through space containing obstacles) rather than saying anything about *how* to do it. Collision free path planning is a well studied problem of robot intelligence, to which different approaches were applied, i.e. Neural Networks, Potential Fields [1, 2], Genetic algorithms [3, 4], Particle Swarm Optimization method [5] and other. It represents at the same time the basic problem of robot intelligence providing convenient model for evaluation of novel approaches in artificial intelligence.

Optimization methods can generally be classified into two distinct groups: direct search and gradient based search. Gradient based search techniques require derivative information of the function and constraints, while direct search methods use only objective function and constraint values. Since derivative information is not used, these methods generally require large number of iterations for convergence, but are at the other hand applicable to very broad problem space.

Swarm Intelligence is an emerging field of Artificial Intelligence. It is concerned with modelling of social interactions between social beings, primarily ants, birds and, in the recent time, bees.

This approach utilizes simple and flexible agents that form a collective intelligence as a group. This is an alternate approach to traditional intelligence models, exhibiting features of

autonomy, emergence, robustness and self-organization. Several examples of artificial models inspired by interactions of social organisms are: Particle swarm optimization (PSO) method, a population based stochastic optimization technique developed in 1995. by Eberhard and Kennedy [6]. It is inspired by flocking behaviour of the birds searching for food. Although PSO methods share many common attributes with GA, such as stochastic nature, population of solution candidates, PSO methods, unlike GA use a kind of cooperation between particles to drive the search process. PSO methods have no evolutionary operators like crossover and mutation. Each particle keeps track of its own best solution, and the best solution found so far by the swarm. It means that particles posses own and collective memory, and are able to communicate. The difference between global best and personal best is used to direct particles in search space.

Another model of swarm-based approaches to optimization is Ant Colony Optimization, (ACO), where the search process is inspired by the collective behaviour of trail deposit and follow-up, observed by real ant colonies. A colony of simple agents (the ants) communicates indirectly via dynamic modifications of their environment (trails of pheromones) and thus proposes solution to a problem, based on their collective experience. Ant colony algorithms as evolutionary optimization algorithms were first proposed by Dorigo (1992) [7] as a multi-agent approach to different combinatorial optimization problems like travelling salesman problem and the quadratic assignment problem. Although, the first results were not very encouraging, it initiated the interest among research community, and since then, several algorithms have been proposed, some of them showing very convincing results [8].

Honey-bees mating may also be considered as a typical swarm-based approach to optimization. It is inspired by behaviour of eusocial insects, which are characterized by three main features, cooperation among adults in brood care and nest construction, overlapping of at least two generations, and reproductive division of labour, respectively. In a recent work, Abbass developed an optimization algorithm based on the honey-bees mating process [9, 10, 11].

This paper presents concept of swarm intelligence, as an optimization technique proposed for finding collision free paths in work area containing different shaped and distributed obstacles. Thus, problem of finding suitable path is considered as an optimization problem, whereat collision free paths have higher fitness value relative to these resulting in collision with an obstacle. This is described in details in section 6 of the paper.

2. STRUCTURE OF A HONEY-BEE COLONY

A honey-bee colony typically consists of a single egg laying queen, usually from zero to several thousands drones and 10000 to 60000 workers [12]. Drones are the fathers of the colony. They are haploid and act to amplify their mother's genome without alteration of their genetic composition except through mutation. Workers specialize in brood care and sometimes lay eggs. Broods arise from either fertilized or unfertilized eggs, whereby the former represent potential queens or workers, and the latter represent prospective drones. The mating process occurs during mating-flights far from the nest. A mating flight starts with the dance where the drones follow the queen and mate with her in the air. In a typical mating-flight, each queen mates with seven to twenty drones. In each mating, sperm reaches the spermatheca and accumulates there to form the genetic pool of the colony. Each time a queen lays fertilized eggs, she retrieves at random a mixture of the sperms accumulated in the spermatheca to fertilize the egg.

3. ARTIFICIAL MODEL

The main steps of the algorithm to be modelled are: mating flight of the queen with the drones, creation of new broods by the queen, improvement of the broods by workers, adaptation of workers fitness, replacement of the queen with the fitter brood. The mating flight may be considered as a set of transitions in a state-space (the environment) where the queen moves between the different states in some speed and mates with the drone encountered at each state probabilistically, according to (1).

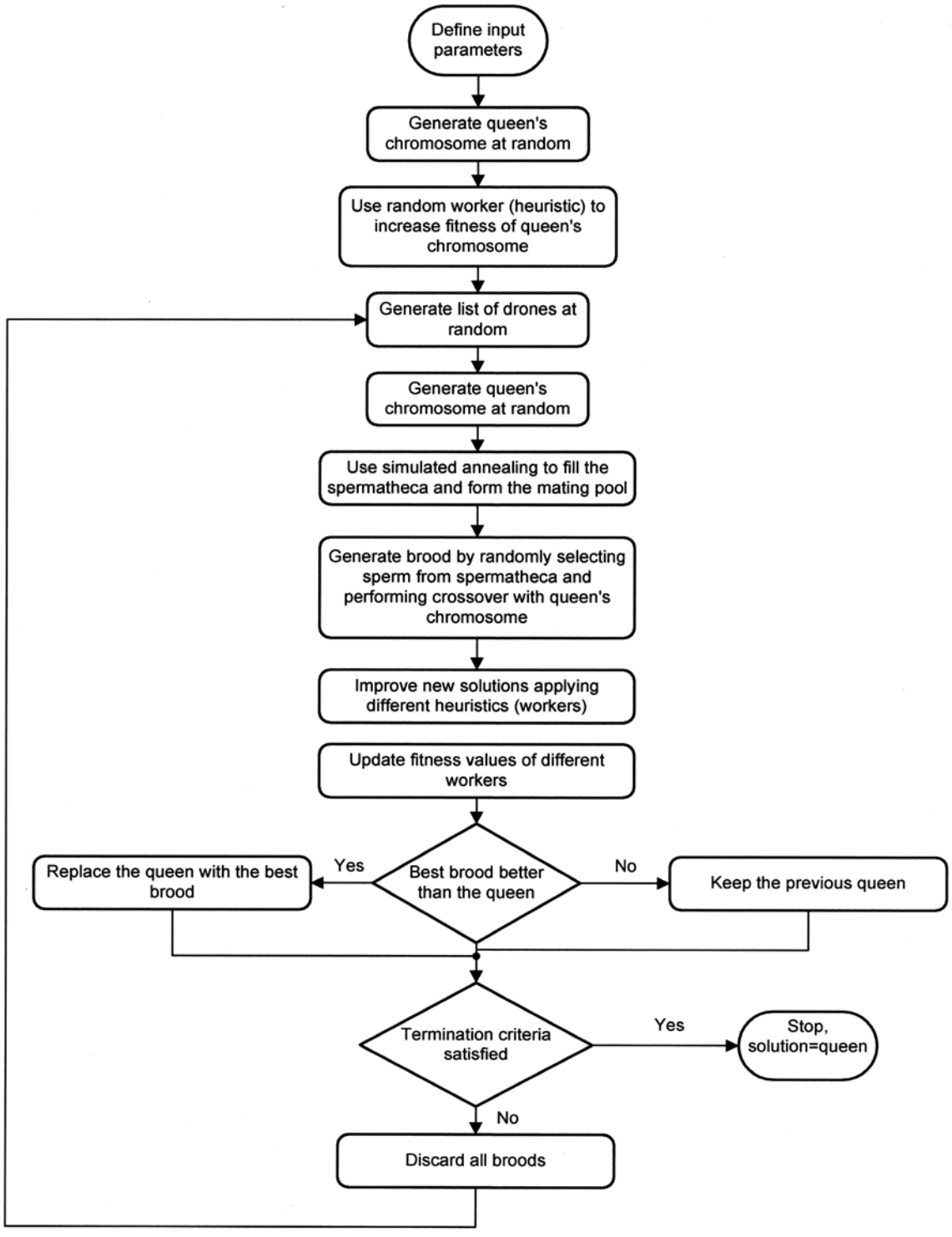

Figure 1: Flowchart of the MBO algorithm.

At the start of the flight, the queen is initialized with some energy content, typically this is a random value from range (0, 1] and returns to her nest when energy content equals to zero or when her spermatheca is full. In developing the algorithm, the functionality of workers is restricted to brood care, and therefore, each worker may be represented as a different heuristic which acts to improve a set of broods.

A drone mates with a queen probabilistically according to annealing function:

$$prob(Q,D) = e^{-\frac{\Delta(f)}{S(t)}} \tag{1}$$

Where prob(Q,D) represents the probability of successful mating, i.e. the probability of fitness of the drone and the queen, and $S(t)$ is the speed of the queen at time t. According to defined annealing function, the probability of mating is high when the queen is at the start of her flight, and therefore, her speed is high, or when the fitness of the new potential drone is similar to the queen's fitness. The main steps of the algorithm are presented in Fig. 1.

After each transition in space, the queen's speed $S(t)$ and energy $E(t)$ decay using the following equations:

$$S(t+1) = \alpha \cdot S(t) \tag{2}$$

$$E(t+1) = E(t) - \gamma \tag{3}$$

where α is a factor in range [0.5, 1] and γ is calculated according to expression:

$$\gamma(t) = \frac{0.5 \cdot E(t)}{M}, \tag{4}$$

where M is the size of spermatheca.

4. ALGORITHM APPLICATION TO DIOPHANTINE EQUATION

In order to perform a test of the algorithm, we applied it to a benchmark Diophantine problem. Diophantine equation is an algebraic equation:

$$f(x_1, x_2, ..., x_n) = 0 \tag{5}$$

which must be solved over the integers $x_i \in Z$. Diophantine problems have a long pedigree in number theory. They also constitute some of the hardest problems in modern mathematics. It is proposed that Diophantine equations make good and difficult benchmark problems for testing different kinds of optimization algorithms [13].

In this paper we will compare behaviour and results of a HBO and GA applied to the Diophantine nonlinear equation of the form, i.e. Markoff equation:

$$x^2 + y^2 + z^2 = 3xyz \tag{6}$$

which has important applications in number theory and known solutions. This example is chosen because it is known how to generate all the solutions in a cube of given size. In the first test case, we will reduce the problem to 2D space by fixing z = 433, to have a unique solution, and finding integers that satisfy:

$$x^2 + y^2 + 433^2 - 1299xy = 0 \tag{7}$$

of which the search space is highly complex in terms of size, see Fig. 2.

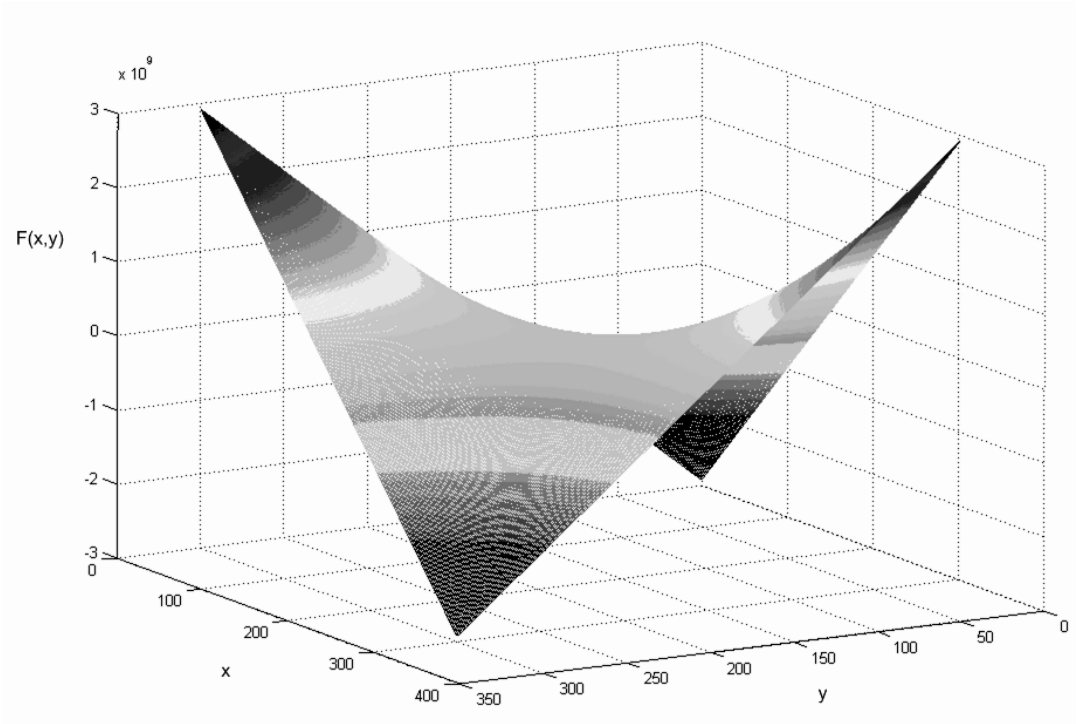

Figure 2: Search space for the first problem.

In the second part of the paper, a HBO will be applied to navigation of a mobile robot through the space with the obstacles in work area.

4.1. Diophantine equation results

HBO and GA algorithm were applied to find solutions of above mentioned problem by searching for values between 0 and 400 for each parameter. Fitness function for this example is very simple and same for GA and HBO.

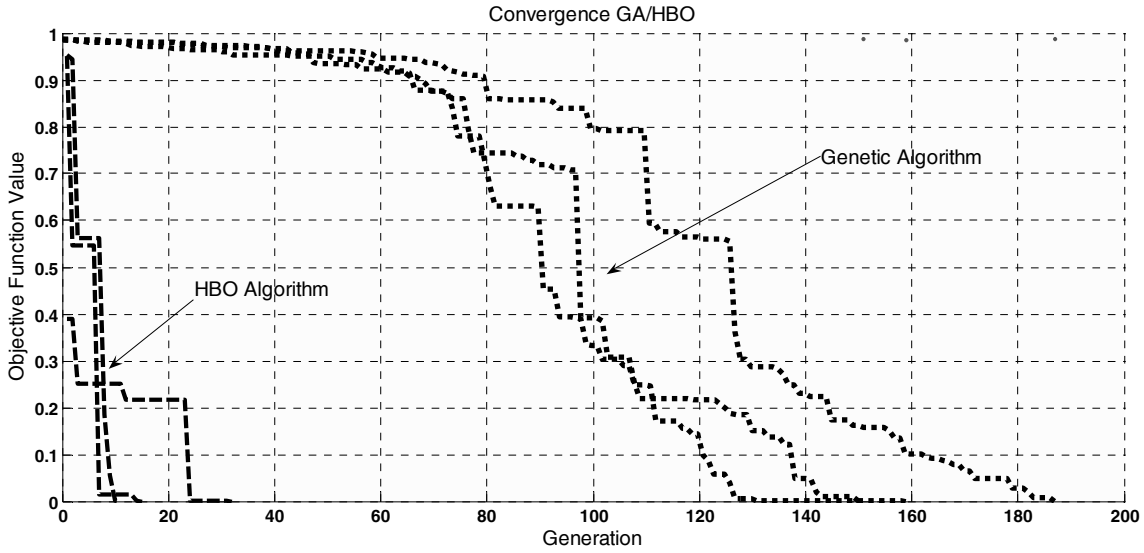

Figure 3: Fitness value for HBO and GA algorithms for the first test case.

It is equal to value of the function defined in (7) and normalized to range [0, 1]. In other words, pairs of numbers which yields lower values of function defined in (7) have higher chances of survival, ideally approaching zero for solution and termination criteria satisfaction.

On this example, both of the algorithms performed well, finding solutions in several hundreds of generations. It is important to notice, that performance of the HBO algorithm depends on the *depth of stochastic search.* We included only two workers, i.e. different heuristics, namely, random walk (RW) and 2-point crossover (2PCO). That means, that in each generation of the main loop, several hundreds local iterations (heuristics) take place for improving the broods. In our examples, depth of the local searches is 100 iterations.

By HBO algorithm, a kind of elitist function is implicitly included, because, the queen is always represented by the best chromosome found so far in all iterations.

By GA on the contrary, we included 10 % elitism, i.e. 10 % of best chromosomes from each generation are copied directly to new generation, thus preserving the best solutions. Without elitism, GA was outperformed by HBO clearly.

Results and behaviour for the HBO and GA are presented by the Fig. 3 for the first test case. Fitness values are normalized, so direct comparison of fitness' values is possible. It is obvious that in case of HBO algorithm, the search starts from the fitness value less than 1, which is a consequence of the first worker applied to initial queen's chromosome.

Table I: GA vs. HBO performance comparison.

	No. of runs	**No. of solutions found**	**Average No. of generations**	**Standard deviation**
GA	30	28	340.0	44.6
HBO	30	30	22.9	4.6

In the case of GA, the search starts from the value near to 1, which represents the best chromosome found in the initial, randomly generated population. We tested each algorithm 30 times and present results of performance for HBO and GA for the first test case in Table I.

Parameters of the GA are: crossover probability: 0.7; mutation probability: 0.01; population size: 30; survival selection: generational; number of offspring: 30; initialization: random, termination condition: solution found or no fitness improvement within last 50 generations. HBO parameters are: spermatheca size: M = 12; stochastic search depth: 100; number of broods: 30; queen's energy E and speed S randomly initialized in range [0.5, 1], energy reduction step $\gamma(t)$ - Eq. (4), heuristics included: random walk and two-point crossover. Termination conditions are: solution found or no queen's fitness improvement over last 50 generations. In 30 runs, GA was twice not able to find satisfying solution, whereas HBO found it in all 30 runs. Although HBO requires fever main loop iterations, it requires actually slightly larger CPU time for finding the solution because of implemented local searches.

5. COLLISION FREE PATH PLANNING RESULTS

In the example that follows, HBO algorithm is implemented to solve the problem of navigation of the mobile robot through the space which contains obstacles. The robot is considered as an object able to move stepwise to front, back, left or right. It has to find the collision free path from an initial point to the destination as formulated in our previous work [14]. We experimented with different space configurations, from just one point-obstacle, to creating maze-like environments and randomly distributed obstacles among the space. In order to evaluate fitness of a trajectory found by robot, we implemented fuzzy fitness / penalty function.

$$f = w_1 \cdot cityblock\left(current\_position - goal\_position\right) + \frac{w_2}{p} \tag{8}$$

Where: f is the fitness of the individual, which has to be minimized (as robot approaches the goal, distance is converging to zero); cityblock is the cityblock function to measure current distance from robot to goal, p is the cityblock distance of an obstacle with which robot has collided from the origin, w_1 and w_2 are weight factors within the range (0, 1].

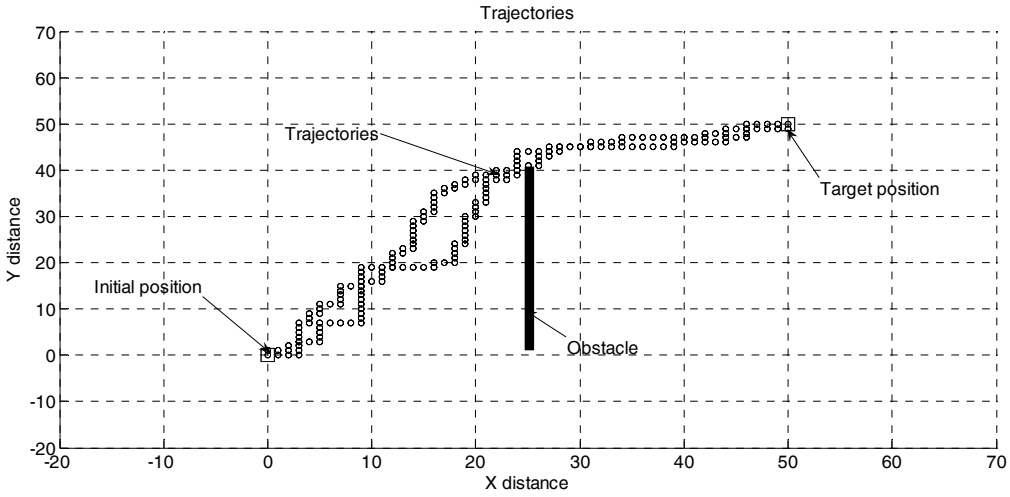

Figure 4: Trajectories found by HBO for simple environment.

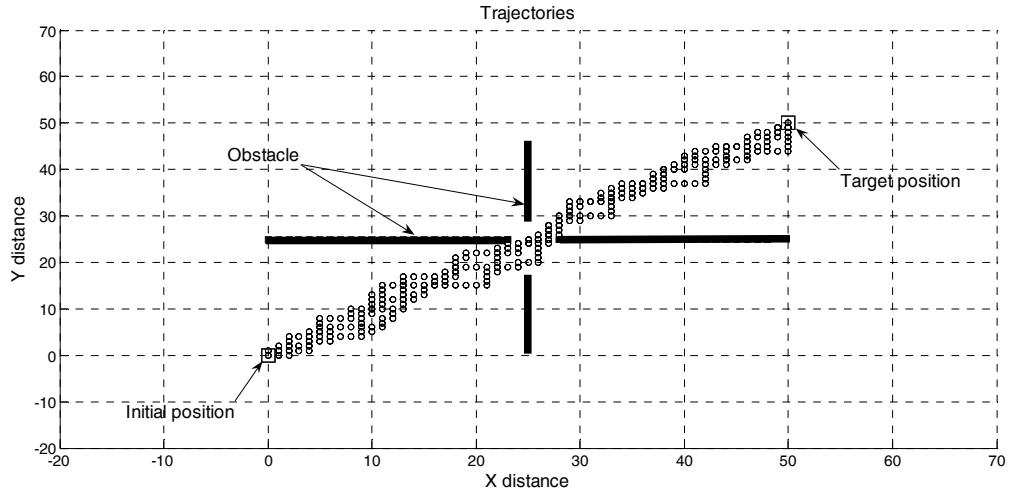

Figure 5: Trajectories found by HBO for environment with four obstacles.

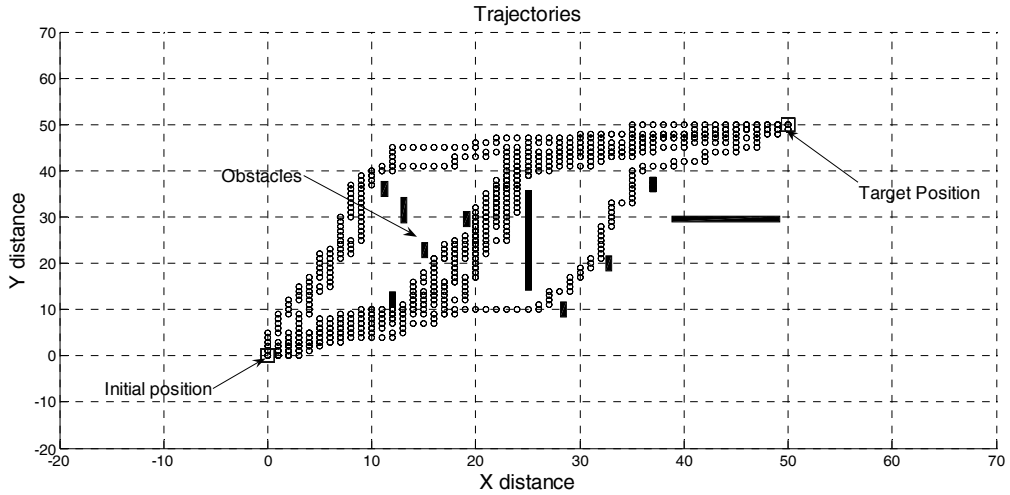

Figure 6: Trajectories found by HBO algorithm for random environment.

During the experiments it was shown that we obtain best performance of the algorithm in terms of number of solutions found with values of $w_1 = 0.01$ and $w_2 = 1$. Using this kind of fitness function, we are able to distinguish paths which collide with an obstacle near to the goal from the paths which collide near the initial position. That way, good parts of good paths can be preserved, and speed of the algorithm is increased. It is known, however, that fine tuning of parameters for GA is not an easy task.

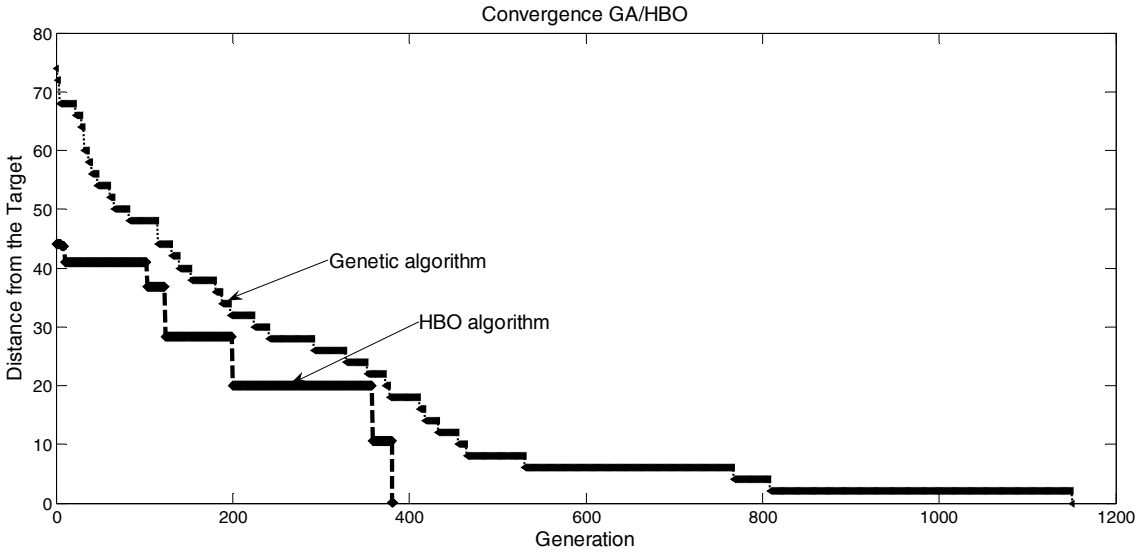

Figure 7: Convergence diagrams for GA and HBO for environment with 10 obstacles (Fig. 4).

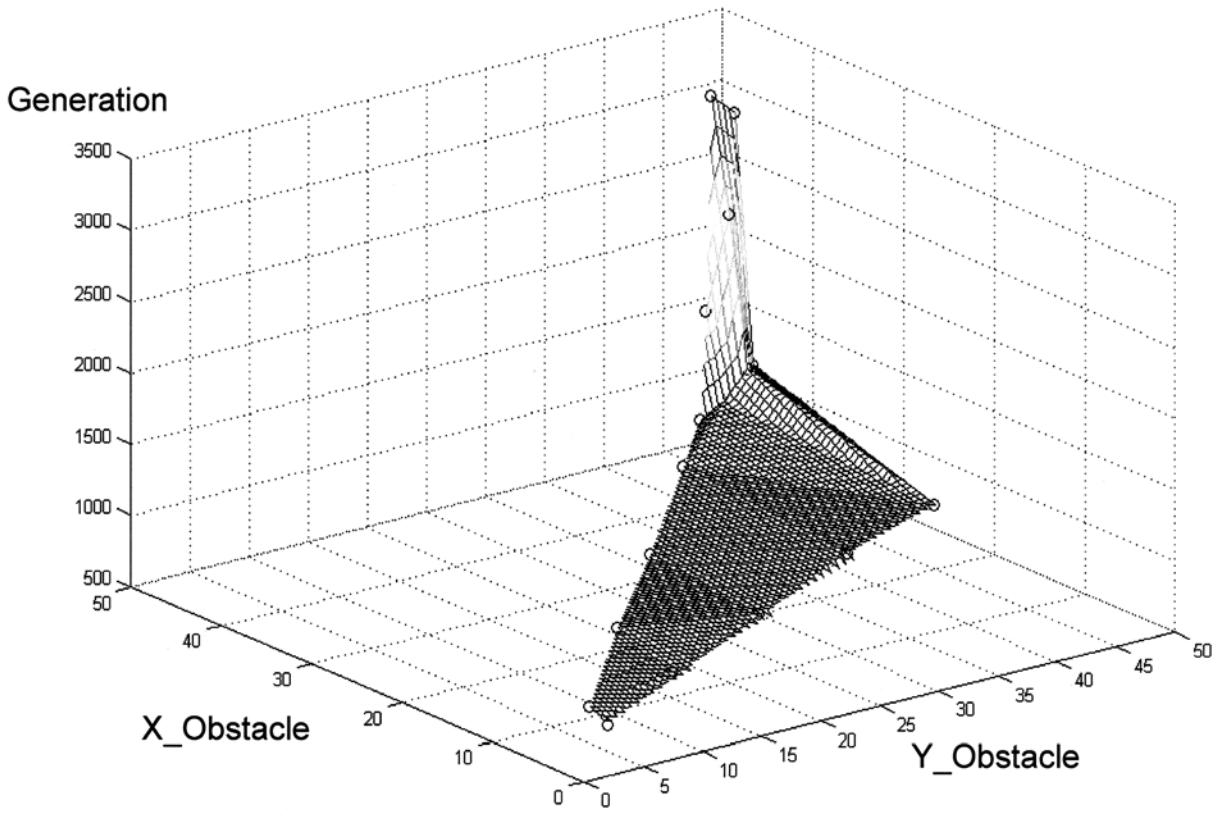

Figure 8: Number of generations in respect of obstacle position.

Robot has reached the goal when the fitness / objective value is equal to 0, i.e. paths which collide with an obstacle and leading robot away from the goal are penalized, and paths which lead robot to the nearest position of the goal by collision free trajectory are preserved.

Fig. 4 illustrates an environment with one large obstacle, occupying middle section of the work space and two trajectories found by the HBO. The algorithm is able to find multiple solutions for one environmental structure. This is due the stochastic nature of the algorithm and no additional operators (such as smoothness of the trajectories for example) included in the fitness function. Fig. 5 presents trajectories found for the setup with four obstacles and free space around the middle of the work space. Fig. 6 illustrates complex environment with ten obstacles, of which positions and sizes are randomly distributed around the work space, but constrained in way not to be distributed in the proximity of start and target positions less then 10 distant units. (Regarding our fitness function defined in (8) for this particular two cases, this could lead to a singular solution). Fig. 7 presents comparison of the convergence diagrams for the GA and HBO for the case of the environment with ten obstacles, see Fig. 6. It is interesting to notice that GA requires on the absolute level about three times more generations compared to HBO, which is similar ratio to the first problem, Diophantine equation. We have to be careful with conclusions though, because severe dissipations of results obtained could occur due to different setups of the algorithm's parameters. We noticed that the performance of the algorithms heavily depends on the number and distribution of obstacles in robot's work space. For obstacles near the middle of the search space, or near the start point, the algorithm is able to quickly find paths which don't collide with obstacle. This is explainable via the Schema Theorem [15], where algorithm quickly finds instances of schemas that avoid obstacles and further easily develops good paths as tails of the chromosomes. For the obstacles near the goal, on the contrary, algorithm requires much more generations for finding good paths, because penalty / fitness function plays role after some time is already spent for developing paths as heads of the chromosomes. This is illustrated with the Fig. 8, where, the x and y axes present position of an obstacle, and z axis number of generations for finding collision free path.

Both algorithms had problems finding solutions for environments with large number of obstacles randomly spread through the space. Algorithms are getting trapped in local optima for such setups. One possible solution of this problem could be implementation of *segmentation* of environment combined with learned experience. Namely, it is a very high level of abstraction to implement finding a path for the whole environment at once. In real problems, robot will very seldom have the chance to see complete work space. Therefore, robot could learn to solve some set of simple group of problems (environments) using HBO or GA. In the second phase, identification of the segmented part of environment would be implemented and appropriate solution found in experience executed.

It is interesting, that HBO actually has a natural mechanism of importing "fresh blood" in the population through the constantly randomly initialized drones, whereas GA has only mutation as a parameter of ensuring appearance of completely new solutions. This could be the reason for HBO to have solutions of higher completeness i.e. more solutions found for the same problem, see Table I. It is possible to include special procedures for GA to constantly ensure new solutions, but this requires extra effort and is not a trivial task.

6. CONCLUSIONS

HBO algorithm is applied to solve a benchmark Diophantine problem and problem of path planning. It is shown that HBO algorithm is comparable to the GA developed for the same purpose in terms of CPU time. The HBO has performed slightly better for Diophantine equation problem, being able to find all 30 solutions for 30 trials, whereby GA found 28

solutions out of 30 trials. In the second part of the work we simulated robot trajectories applying HBO and GA for the same purpose.

Proposed algorithm enables robot to self-organize in complex environment, based solely on information about what is wanted to do. There is no knowledge about how to do it. Here we also propose fuzzy fitness function for evaluation of the trajectories. The results of algorithms are comparable here as well, with CPU time slightly on the side of the GA, but again, HBO was able to find more collision free paths on the same set of trials for complex environments. In our work only two heuristics (workers) for local search were implemented in HBO, whereat RW outperformed 2PCO strongly.

The authors expect to achieve better results of HBO in terms of CPU time and number of successful solutions by implementing different heuristics, such as GSAT for example. Authors are also considering possibilities of different encoding principles, modifying fitness function to be able to evaluate smoothness, adding a member of fitness equation for evaluation of quality according to aberration of an ideal trajectory and implementing the algorithm to real Pioneer 2DX robot.

Future investigation will also include implementation of evolvable parameters for the algorithms and possibilities of forming experience from learned trajectories. Another direction of future investigation will be application of environmental segmentation with aim of subdividing the work space to simpler groups of problems.

7. ACKNOWLEDGEMENTS

Authors would like to acknowledge support of Croatian Ministry of Science, Education and Sports, through projects No.: 0120015, Intelligent Automated Assembly, 01201201948-1941, Multi-agent Automated Assembly and joint technological project TP-E-46, with EGO-Elektrokontakt d.d.

REFERENCES

[1] Latombe, J. C. (1991). *Robot motion planning*, Kluwer Academic Publishers, Boston

[2] Bekey, G. A.; Goldberg, K. Y. (1993). *Neural networks in robotics*, Kluwer Academic Publishers, Boston

[3] Xiao, J.; Michalewicz, Z.; Zhang, L.; Trojanowsky, K. (1997). Adaptive evolutinary planner/navigator for mobile robots, *IEEE Transactions on Evolutionary Computation*, Vol. 1, 18-28

[4] Castillo, O.; Trujillo, L. (2005). Multiple objective optimization genetic algorithms for path planning in autonomous mobile robots, *International journal of computers, systems and signals*, Vol. 6, No. 1, 48-63

[5] Chen, M.; Yangmin, L. (2006) Smooth formation navigation of multiple mobile robots for avoiding moving obstacles, *International Journal of Control, Automation and Systems*, Vol. 4, No. 4, 466-479

[6] Eberhart, R. C.; Kennedy, J. (1995). A new optimizer using particle swarm theory, *Proceedings of the Sixth International Symposium on Micromachine and Human Science*, Nagoya, Japan, 39-43

[7] Colorni, A.; Dorigo, M. & Maniezzo, V. (1992). Distributed optimization by ant colonies, *Proceedings of the First European Conference on Artificial Life*, Paris, France, F. Varela and P. Bourgine (Eds.), Elsevier Publishing, 134-142

[8] Dreo, J; Petrowsky, A; Siarry, P; Taillard, E. (2006). *Metaheuristics for hard optimization*, Springer Verlag, Berlin

[9] Abass, H. A. (2001). A single queen single worker honey bees approach to 3-sat, *The genetic and Evolutionary Computation Conference*, GECCO2001, San Francisco, USA

[10] Abass, H. A. (2001). Marriage in honey bees' optimization: A haplometrics polygonus swarming approach, *The Congress on Evolutionary Computation*, CEC2001, Seoul, Korea
[11] Abass, H. A. (2002). An agent based approach to 3-sat using marriage in honey-bees optimization, *International Journal of Knowledge-based Intelligent Engineering Systems*, Vol. 6, No. 2
[12] Laidlaw, H.; Page, R. E. (1986). *Bee genetics and breeding*, pages 3-22, Academic Press Inc
[13] Bull, P.; Knowles, A.; Tedesco, G. (2006). Diophantine benchmarks for the b-cell algorithm, *International Conference on Artificial Immune Systems*, Canterbury, Great Britain
[14] Ćurković, P.; Jerbić, B.; Vranješ, B. (2006). Genetic algorithm for robot path estimation, *Proceedings of the 17th International DAAAM Conference on Intelligent Manufacturing and Automation*, Vienna
[15] Holland, J. H. (1975). *Adaption in natural and artificial systems*, University of Michigan Press, Ann Arbor