Intelligent Control and Evolutionary Strategies Applied to Multirobotic Systems

Gustavo Pessin, Fernando Osório, Alberto Y. Hata and Denis F. Wolf
Institute of Mathematics and Computer Science (ICMC)
University of São Paulo (USP)
Av. do Trabalhador São-Carlense, 400 – C.P. 668 – 13.560-970 – São Carlos, SP, Brazil
Email: {pessin,fosorio,alberto,denis}@icmc.usp.br

***Abstract*—This paper describes the modeling, implementation, and evaluation of RoBombeiros[1] multirobotic system. The robotic task in this paper is performed over a natural disaster, simulated as a forest fire. The simulator supports several features to allow realistic simulation, like irregular terrains, natural processes (e.g. fire, wind) and physical constraint in the creation and application of mobile robots. The proposed system relies on two steps: (i) group formation planning and (ii) intelligent techniques to perform robots navigation for fire fighting. For planning, we used genetic algorithms to evolve positioning strategies for firefighting robots performance. For robots operation, physically simulated fire-fighting robots were built, and the sensory information of each robot (e.g. GPS, compass, sonar) was used in the input of an artificial neural network (ANN). The ANN controls the vehicle (robot) actuators and allows navigation with obstacle avoidance. Simulation results show that the ANN satisfactorily controls the mobile robots; the genetic algorithm adequately configures the fire fighting strategy and the proposed multi-robotic system can have an essential hole in the planning and execution of fire fighting in real forests.**

I. Introduction

Like in nature, there are many fields where a single agent is not sufficient or efficient to fulfill a task. Tasks like cleaning nuclear residuals, cleaning chemical accidents, forest fire combat or even on constructions, agriculture, hostile environment exploration, security and critical missions may be better accomplished using a group of agents. The use of robotic agents instead of human beings, in these applications, may add security, reliability and efficiency. The multirobotic systems are extremely dependent on control techniques; they can add mobility, flexibility and robustness of a new way to a wide range of new applications [1], but bring a series of new questions to be solved in collaboration and cooperation. Specialized algorithms, composed by rules and automats have been developed seeking to coordinate these physical sets in dynamic environments, showing to be an extremely complex challenge [2]; due to it, a great number of researchers are migrating their efforts to several different approaches (e.g. application of classical intelligent artificial techniques, social models, market base models, swarm based models).

Navigation is a fundamental problem of robotics. Traversing from a place to other depends of three fundamental aspects: localization, orientation and motion controlling. To know both localization and orientation, the mobile robot must hold sensors (e.g. GPS, compass). For motion control, it must have an appropriate number of motors. Sensors and actuators normally are subjected to errors and interferences, thus the robot action control must always take into account the imprecision of involved sensors and motors. A robust system must allow, even with imprecise sensors and actuators, the robot to establish its objective. One machine learning technique appropriate for this task are artificial neural networks, given its capability of learning from examples, and generalization and adaptation of the outputs. This is a method largely used in reactive systems navigation controlling [3].

In the firefighting mission, one of the most important questions is related to the robot position setting. According to the actuation capacity of each robot, the weather condition (wind, rain), the topography, and the vegetation, varied arrangements can be proposed. These arrangements, when suggested by a specialist, may not take in account a large number of variables, in this manner, for this task we can use machine learning techniques. A machine learning method recommended for these cases is genetic algorithms (GA) [4], [5], which consists in a global optimization algorithm that employs parallel and structured search strategies, directed to fitness points seeking. Allowing the multi-criteria search in a multidimensional space and being unsupervised, don't make necessary any previous information database.

In [6] we describe a simulation environment for wildfires identification and combat with robots controlled only by rules. In this paper we present the evolution of that work, which has been redesigned using the physical simulation library Open Dynamics Engine [7] and where the navigation and the formation of the group count with machine learning techniques.

This paper has the following structure: Section II introduces short theoretical description of robot's applications. Section III presents concepts of machine learning. In section IV we explains the developed environment. Section V describes the multirobot operation for identification and firefighting, the robot morphology, the building and evaluation of artificial neural network and the building and evaluation of genetic algorithms performing group formation. We finalize presenting the conclusion of the presented work and the future perspectives.

[1]Source-code and videos availables at http://pessin.googlepages.com

II. Mobile Robotics

The problem-solving task capabilities of multi-robotics system depend on higher developed capacities of each single robot; they can count including with different robots capacities (such as heterogeneous systems). Several current works demonstrates mobile robotic usage (individual system) on hostile operations as the rescue auxiliary robot Raposa [8] and SACI robot [9] designed for acting on fire combat. The militaries prototypes Boeing X-45 [10] and nEUROn [11] that, under human-landed supervising (without embedded pilot) are being tested for combat missions. Moreover, there are robots to perform tasks on aquatic environments, to space, caves and volcanoes exploration, and even to household use. There are also competitions [12], [13] that uses small autonomous mobile robots that have missions like find and put out a candle (as a simulated fire).

Multirobotic systems must be formed by robots that are able to effective act on tasks, so knowledge about robotic control is a very important field. Works describing intelligent robot navigation can be seen in [14], [3]. In 2004 and 2005, DARPA Grand Challenge [15], financed by the Defense Advanced Research Projects Agency organized a competition where the goal was building a completely autonomous vehicle that could complete a long way on dirt road on limited time. In 2007 the focus of the competition has changed. Renamed to DARPA Urban Challenge, had a new goal to build a vehicle that could be autonomous on urban traffic, and realize tasks like parking, overtaking and intersection negotiations. These examples show trends in cooperation and multiple interactions.

The work with groups adds a great number of possibilities on tasking-solving but bring a series of new questions to be solved in collaboration and cooperation. Works using multi-robotic systems like [16], [17] uses pre-programmed rules on agents to perform formation. In [1], [18] are explored techniques to perform works with collectives robotics, used mainly for the purpose of applying the concept of self-organization and collective optimization, but task division is not directed explored. The works described demonstrate that the application of mobile robotics in control of incidents is an important and active topic of research and development. These several competitions also demonstrate that there isn't a definitive or more adequate solution to the problem, and it's an open research field. In all consulted documents there's no consensual form to multirobotic system's conformation and actuation. Unpredicted situations with large degree of autonomy and robustness are still difficult to handle.

III. Machine Learning

Machine learning is an artificial intelligence field that has with objective develop computational learning and knowledge acquisition techniques [19]. These techniques try to achieve an intelligent behavior and perform complex tasks with competence level equal or higher than a human specialist [20]. To build the system proposed in this paper we used neural networks and genetic algorithms, in this way, we described briefly its features below.

Artificial neural network (ANN) are universal approximators that make mappings in multi-variable function space [21]. The capacity of learning and generalization of ANNs are one of the greatest attractive. The information processing in an ANN is done by artificial neural structures [19]. These structures, as well as the artificial neuron itself are biological analogies to the brain behavior. The backpropagation algorithm [22] is a supervised learning algorithm of ANNs. The learning training basis is a set of data that must present for each input, the provided output for the system. The ANN training must involve several simulation runs, initializing the weights randomly. Other important question is the generalization level that is usually measured through a validation base used in parallel to the training base.

Genetic algorithms (GA) [4], [5] are global optimization techniques that employ parallel and structured search strategies. Allow the multi-criteria search in a multi-dimensional space. They are methods classified as unsupervised, being unnecessary any previous information database. The GAs use iterative procedures that simulate the evolution process of a population constituted by candidate solutions of a certain problem. The evolution process is guided by a selection mechanism based on fitness of individual structures. For each algorithm iteration (single generation), a new structure set is created by information changing (bit or blocks) between selected structures of the previous generation [23]. The result of this conduces to the increasing in the individual fitness. A GA is structured in a way that the information about a determined system can be coded similarly to a biological chromosome, like a value array, where usually each sequence fragment represents a variable.

When dealing with applications, [24] uses a GA to satisfactorily optimize trajectory planning for a robot arm. In [25], a GA model correctly evolves values for force and time application to allow a robot to walk. The work [26] presents a GA model to evolve the exploration method of a mobile robot in an unknown environment. These works presents acceptable results for static environments; [27] describes a possible solution for operation in dynamic places, where the robot perform the navigation using GA. This robot is equipped with obstacle sensors and when identifies a possible collision, it stops and executes again the planning module using GA. In this way, the system becomes suitable for dynamic environments.

IV. Prototype implementation

In order to better understand how to proceed in wildfires combat, and plan the strategies to be implemented in mobile robots, we elaborated a study about real operation techniques. This study was based on works of [28], [29], [30], [31]. To implement the fire spreading, we obtained by [32] real velocity measurements. Research about forest models and residues is very important to improve the simulation models to be reproduced in virtual environments [6]. For vegetation simulation and correct fire propagation, there is a hidden matrix under the terrain. This matrix has for each terrain region, the type of present vegetation, consequently, associating this information with the wind orientation and intensity we can build the fire propagation simulation. Regarding the wind, both its intensity and its orientation can be generated randomly or configured with parameters defined by the user. The time of permanence of the fire in an area is related directly to the present vegetation type and behaves in basis of terrain type values, terrain slope, wind orientation and intensity. In this way the fire spreading simulation try to model as realistic as possible the fire propagation in the real environment. The detailed characteristics about fire spreading modeled to this work, as well as the forest fuel models and the real operation techniques are compiled into [33]. The simulated terrain is based on topographical maps and on forest fuel maps models that can also be seen in [33].

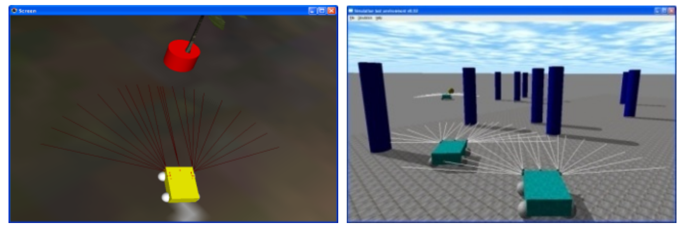
The proposed operation model depends essentially of two steps, *planning* and *action*, so it were implemented specific prototypes for each step. The prototype where is executed the action of fire-fighting robots, controlled by artificial neural network can be seen in Fig. 3 and 4. This tridimensional prototype uses the OSG library [34] which is responsible by graphic output, the Demeter library [35] that is responsible by irregular terrain generation and ODE library [7] which is responsible by physic realism, both in the robotic morphology as in the collision involving the objects presents in the environment (e.g. robots, trees, terrain inclination). Using ODE library allows the physically simulated robots to respect questions like gravity, inertia and friction. The prototype where is accomplished the planning uses a genetic algorithm and does not needs visualization, however it is implemented with the possibility of a 2D graphic output developed in SDL [36], as shows Fig. 3(b). The integration between the prototypes is done by a text file, after the evolution, the responsible prototype builds a file with operations positions that is read in the action prototype initialization.

V. Identification and Combat Operation

We have used the environment to simulate the following operation: a monitor agent checks the forest terrain, when identifies an area with fire focus, it activates the strategy evolution module (detailed in the Subsection V-B). After getting the operation coordinates through GA, the monitor agent sends messages to the combat robots informing its operation positions. The agent behavior is hybrid (it has a plan to execute the navigation and also has a sensorial-motor reaction system), moving in direction to its specific target and avoiding obstacles. For simulated fire-fighting we used the indirect method [31]. The simulated fire-fighting robots are graders that have as purpose surround the fire and create firebreaks (area without vegetation where the fire put out due to lack of fuel). This operation can be understood with the Fig. 3.

A. Mobile Robots Morphology

The mobile robots were developed with the ODE library that supports the simulation of articulated rigid bodies. The Fig. 1 shows the developed vehicle. Given the existence of physic restrictions, the only way to control this vehicle is by applying forces in its two simulated motors that are: an angular motor (for steering wheel spinning) and a linear motor (for torque). Besides the GPS, responsible to localization obtaining, each robot has also a compass, necessary for vehicle orientation achievement.



(a) (b)

Fig. 1. Developed mobile robots with distance sensors.

The azimuth (target angle) is obtained from current position (GPS) and from target position (received by message). The distance sensors are simulated sonar, bringing the features of capacity to measure distance between 15cm and 11m, like Polaroid 6500 (www.senscomp.com).

B. Positioning Evolution Strategy

The planning mechanism uses a GA to define the initial and final operation positions of each robot for fire-fighting, which is developed using GAlib library [37]. Considering that the combat agents are graders which have the finality to create firebreaks, we require that the GA returns the following information: initial and final angle, and initial and final radius for each robot, both related to the fire starting point. The proposed chromosome can be seen in the Tab. I. In this, it is presented information of all group of involved agents, thus, the chromosome size depends on the number of robots in the system. We executed simulations considering the existence of 4 combat agents.

The coordinates of operation are calculated applying Eq. 1 and 2 to the chromosome. Where (x_f, y_f) is the robot's final position, (x_a, y_a) is the starting position of the fire, r_i is the radius (gene 5 to 9) and a_i is the angle (gene 0 to 4). The radius and the angle are specifics to each operation of each robot (initial and final coordinate of firebreaks creation).

TABLE I

Chromosome structure (group of four robots).

Gene	Function	Min. value	Max. value
0	Initial angle of robot 0	0.0º	360.0º
1	Final angle of robot 0; initial of robot 1	0.0º	360.0º
2	Final angle of robot 1; initial of robot 2	0.0º	360.0º
3	Final angle of robot 2; initial of robot 3	0.0º	360.0º
4	Final angle of robot 3	0.0º	360.0º
5	Initial radius of robot 0	10.0m	100.0m
6	Final radius of robot 0; initial of robot 1	10.0m	100.0m
7	Final radius of robot 1; initial of robot 2	10.0m	100.0m
8	Final radius of robot 2; initial of robot 3	10.0m	100.0m
9	Final radius of robot 3	10.0m	100.0m

$$x_f = x_a + r_i \cdot \cos(a_i) \qquad (1)$$

$$y_f = y_a + r_i \cdot \sin(a_i) \qquad (2)$$

Regarding the GAs parameters, we use overlapping populations model proposed by [38] and alleles that limit value set generated for each attribute (radius between 10 and 100 and angles between 0 and 360 degrees). The use of alleles reduces the search space. Also, we used real genome, optimized for float point numbers. Comparative analysis involving binary and float point representations showed that floating point representations has significally advantages principally related to precision and convergence speed [39], [40]. Selection scheme used was stochastic remainder sampling selector that has a better performance compared with roulette scheme [39].

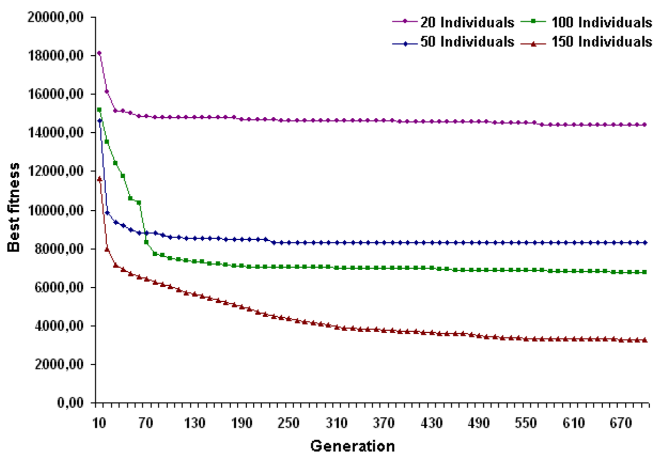
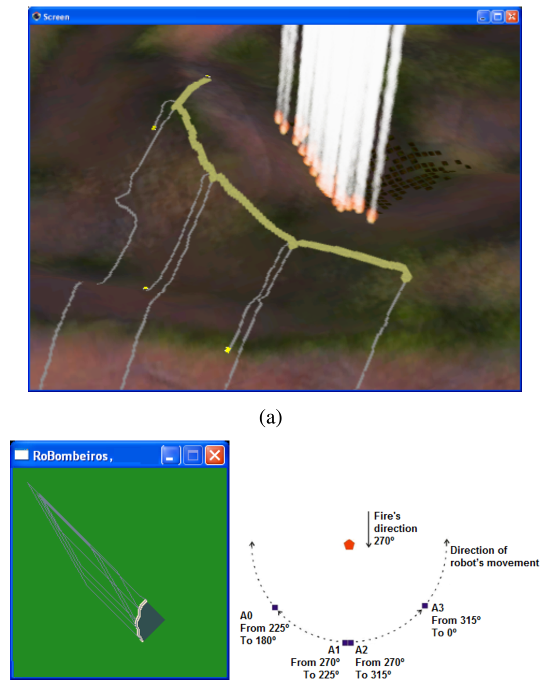


Fig. 2. Evolution of fitness according to number of generations and individuals (group of four robots).

TABLE II
BEST CHROMOSOMES (RESULTANT OF THREE SIMULATION).

	Simulation		
Gene	A	B	C
0	225.72	224.89	134.82
1	199.26	200.77	142.65
2	174.12	177.86	172.33
3	155.20	160.68	196.14
4	136.38	136.64	244.84
5	27.35	28.05	54.94
6	30.45	29.75	42.95
7	33.94	30.58	35.84
8	38.08	31.12	30.37
9	35.96	33.24	26.33

The developed fitness accumulates the following final values for each simulation: (i) Amount of burned area: seeks for burned area minimization; (ii) Amount of area with firebreak: seeks minimize the robots working area, and; (iii) Absolute average error: seeks minimize the difference between the overall average of useful firebreaks related to the useful firebreak created by each agent, so, the working area tends to equalize. In the simulations we looked for fitness value minimization. The graphic for fitness evolution for 4 robots can be seen in the Fig. 2, presenting the average for 3 simulations. We can verify the fitness reduction by the generations. The simulation with 150 individuals was the one that reached the lower fitness. The satisfactory result of this simulation can be checked in the Figs. 3(a) and 3(b). Resulting chromosomes of simulation using 700 generations and 150 individuals is described in the Tab. II.



(a)

(b) (c)

Fig. 3. (a) Planning result using GA with 150 individuals and 700 generations, (b) Same result in 2D visualization and (c) Theoretical sample.

Experiments with groups of 2 robots were also performed, with chromosome adaptation (Tab. I presents the chromosome structure for 4 robots). Both using Gaussian as Uniform mutation we obtained 100% satisfactory results. It was possible to notice the maximum, average and minimum fitness convergence for 2 and 4 robots cases using 150 individuals and, at least, using 400 generations. At all we executed 20 simulations for each parameter set; for the suggested GA parameters, 100% of simulations were able to cover the fire in a satisfactory way. It is important to mention that we has conduced some simulations using different navigation velocity and fire spreading, although, with fewer experiments. These experiments are not detailed in this paper, but also present satisfactory results. More details can be seen in [33].

C. Navigation Control

At the beginning of combat operation, each robot receives a message of type *displace autonomously to (x,y)*. It makes each robot to start navigating in direction to a requested position; being controlled only by the ANN. The intelligent navigation control was developed with an multilayer perceptron ANN, trained with resilent bakpropagation learning algorithm. This

ANN was developed and trained using SNNS [41]. The intelligent control performed by the ANN allows the navigation and obstacle avoidance in a dynamic environment, using only the locally available information obtained by the sensors of mobile robots.

The ANN has as input the following information: (i) Vehicle orientation acquired from a simulated compass; (ii) Vehicle azimuth, obtained using the GPS and the message containing the target coordinate; (iii) Five distance sensor values (sonar). The ANN outputs are: (i) Force to be applied in the angular motor (steering wheel, from -1.5 to 1.5); (ii) Force to be applied on the linear motor (torque, from 0.0 to 6.0). Initial experiments show that it is indispensable that the vehicle decreases the speed in curves, mainly when it is closed. We used a single ANN that controls both the steering wheel as the torque, being able to guide the vehicle and navigate without the necessity of a human control or a previously coded automaton that instruct when must deviate an obstacle or how to navigate. The final database obtained presented 4.985 registries, being divided in 70% for training and 30% for validation.

Once obtained data for training and validation, we started the ANN topology definition. We tested 6 different ANN topologies, with 4, 9, 18, 24, 30 e 36 neurons in the hidden layer. The analysis and selection of the best ANN was made by the Mean Absolute Error (MAE) and the Mean Squared Error (MSE). We executed 3 trainings for each ANN topology in order to analyze the MAE, using a different random seed in each train. The MAE used in this validation was obtained from results analysis of ANN tests with 5.000, 10.000, 20.000, 40.000, 60.000, 80.000 and 100.000 cycles. The ANN with 4 and 9 neurons in the hidden layer not presented learning capacity, however, the ANN training with 18, 24, 30 and 36 neurons in the hidden layer showed that all of these ANN presents learning capacity, being the ANN with 24 neurons in the hidden layer the one that showed the smaller error. In the training cycle with the number 32.500 occurs the training and test error curve inversion (Generalization Optimum Point). This was the chosen ANN to be applied in the simulator. After implementing the selected ANN in the mobile robot control and inserting in the control system the ANN code, we evaluated if the control provided by the ANN is efficient to perform the traversing between the initial and final points given by the monitor agent. In this way, we made numerous experiments with different terrain topologies. We also measured the error (imprecision) in the sensors and actuators supported by the ANN, maintaining the navigation correct. Given the initial and final points for an agent group, the navigation system developed for the environment can be seen in the Fig. 4. The environment was parameterized with 4 occupancy level for trees. We calculated the total occupation area of trees related to the terrain area. The occupation is approximately 10%; 5%; 2.5% and 0.625%. Results are considered satisfactory in the following form: each vehicle must be able to traverse a simulated region that represents an area of approximately 1 km using different tree occupation rates. Vehicles shouldn't collide with trees neither with other vehicles.

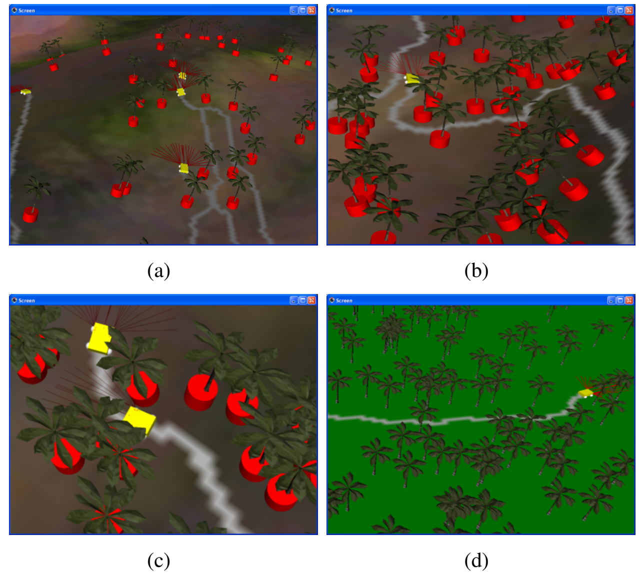


Fig. 4. Paths generated by simulation using an ANN: (a) 2.5% occupation; (b), (c) and (d) 10% of occupation.

The simulation results, considering different types of occupation with trees can be seen in the Tab. III. We can see, in this Table, that for environments with 5% of occupation, or fewer, the ANN was able to perform the navigation with obstacle avoidance in 98% and 100% of the cases. This error occurs when the vehicles entry in vegetated bottlenecking regions; a preliminary way to treat this error would be the attribution of a reverse gear to the vehicle, that isn't yet implemented in our model.

TABLE III
NAVIGATION RESULTS USING THE ANN.

Number of navigation simulation	Area covered with trees	Satisfactory results using the ANN
50	10.00%	42 (84.00%)
50	5.000%	49 (98.00%)
50	2.500%	49 (98.00%)
50	0.625%	50 (100.0%)

Other experiments were executed with the application of a noise in the vehicle sensors and actuators. The noise application of up to 10% maintained the network functionality, but the noise application of 20% presented serious failures. More details can be seen in [33].

VI. Future Works

Some approaches are planed as future work: (i) evaluation of fault-tolerance methods for the proposed operation; (ii) comparing GA with other group formation techniques, like swarm based models [1], [18] and Market-based Approaches [42]; (iii) comparing ANN with other navigation techniques, possibly using map characteristics; (iv) sophistication of the fire simulation model and the robotic actuation. After analyzing these new approaches, the system must be built using real robots.

VII. Conclusion

This paper presented the modeling, the implementation, and the evaluation of a multirobotic system. The robotic task was performed over a natural disaster, simulated as a forest fire. The simulator supports several features to allow realistic simulation, like irregular terrains, natural processes (e.g. fire, wind) and physical restrictions. The proposed system relies on two steps: (i) planning, for group formation and (ii) intelligent techniques to perform robots navigation for fire fighting. For planning, we used genetic algorithms to evolve positioning strategies for firefighting robots performance. For robots operation, physically simulated fire-fighting robots were built, and the sensory information of each robot (e.g. GPS, compass, sonar) was used in the input of an artificial neural network (ANN). The ANN controls the vehicle (robot) actuators and allows navigation with obstacle avoidance. Simulation results show that the ANN satisfactorily controls the mobile robots; the genetic algorithm adequately configures the fire fighting strategy and the proposed multi-robotic system can have an essential hole in the planning and execution of fire fighting in real forests.

Acknowledgment

The authors acknowledge the support granted by CNPq and FAPESP to the INCT-SEC (National Institute of Science and Technology – Critical Embedded Systems – Brazil), processes 573963/2008-8 and 08/57870-9. Also acknowledge Capes for financial support.

References

[1] F. Mondada, L. M. Gambardella, D. Floreano, and M. Dorigo, "The cooperation of swarm-bots: physical interactions in collective robotics," *IEEE Robotics and Automation Magazine*, vol. 12, pp. 21–28, 2005.

[2] J. Go, B. Browning, and M. Veloso, "Accurate and flexible simulation for dynamic, vision-centric robots," *Autonomous Agents and Multiagent Systems, International Joint Conference on*, vol. 3, pp. 1388–1389, 2004.

[3] M. Heinen, F. Osório, F. Heinen, and C. Kelber, "Seva3d: Using arti cial neural networks to autonomous vehicle parking control," in *Neural Networks, 2006. IJCNN '06. International Joint Conference on*, 2006.

[4] J. Holland, *Adaptation in Natural and Artificial Systems*. Ann Arbor, Michigan, USA: The University of Michigan Press, 1975.

[5] M. Mitchell, *An Introduction to Genetic Algorithms*. MIT Press, 1996.

[6] G. Pessin, F. Osório, S. Musse, V. Nonnenmacher, and S. Ferreira, "Simulação virtual de agentes autônomos para a identificação e controle de incêndios em reservas naturais," in *Proceedings of IX Symposium on Virtual and Augmented Reality*, 2007, pp. 236–245.

[7] R. Smith, "Open dynamics engine," Feb. 2007. [Online]. Available: www.ode.org

[8] IdMind, "Projecto raposa," Sep. 2007. [Online]. Available: raposa.idmind.pt

[9] A. M. Macedo, A. L. Macedo, and J. B. F. Duarte, "Robótica aplicada ao combate a incidentes," *TN Petróleo*, no. 53, pp. 108–113, 2007.

[10] Boeing, "Boing company website," May 2007. [Online]. Available: www.boeing.com

[11] Dassault, "Dassault company website," May 2007. [Online]. Available: www.dassault-aviation.com

[12] Trinity, "Firefighting robot contest," Sep. 2007. [Online]. Available: www.trincoll.edu/events/robot

[13] PennState, "Mobile robotics program," Jun. 2007. [Online]. Available: www.ecsel.psu.edu

[14] Y. Zhao and E. G. Collins, "Robust automatic parallel parking in tight spaces via fuzzy logic," *Robotics and Autonomous Systems*, vol. 51, pp. 111–127, 2005.

[15] Darpa, "Grand challenge website," May 2007. [Online]. Available: www.darpa.mil/grandchallenge

[16] H. Yamaguchi, "Adaptive formation control for distributed autonomous mobile robot groups," in *Robotics and Automation, 1997. Proceedings., 1997 IEEE International Conference on*, vol. 3, Apr 1997, pp. 2300–2305 vol.3.

[17] T. Balch and R. C. Arkin, "Behavior-based formation control for multirobot teams," *IEEE Transactions on Robotics and Automation*, vol. 14, no. 6, pp. 926–939, 1998.

[18] M. Dorigo, V. Trianni, E. Sahin, R. Gro, T. Labella, G. Baldassarre, S. Nolfi, J. Deneubourg, F. Mondada, D. Floreano, and L. Gambardella, "Evolving self-organizing behaviors for a swarm-bot," *Autonomous Robots*, vol. 17, pp. 223–245, 2004.

[19] S. O. Rezende, *Sistemas Inteligentes: Fundamentos e Aplicações*. Barueri, SP: Manole, 2003.

[20] C. Nikolopoulos, *Expert Systems: Introduction to First and Second Generation and Hybrid Knowledge Based Systems*. New York, USA: Marcel Dekker Inc. Press, 1997.

[21] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Netw.*, vol. 2, no. 5, pp. 359–366, 1989.

[22] D. Rumelhart and J. McClelland, *Parallel Distributed Processing*. Cambridge: The MIT Press, 1986.

[23] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Professional, 1989.

[24] V. B. Nguyen and A. S. Morris, "Genetic algorithm tuned fuzzy logic controller for a robot arm with two-link flexibility and two-joint elasticity," *Journal of Intelligent and Robotic Systems*, vol. 49, no. 1, pp. 3–18, 2007.

[25] M. R. Heinen and F. S. Osório, "Applying genetic algorithms to control gait of simulated robots," *(CERMA) Electronics, Robotics and Automotive Mechanics Conference*, pp. 500–505, 2007.

[26] Y. Zhou, "An area exploration strategy evolved by genetic algorithm," Master's thesis, University of Georgia, 2005.

[27] H. Burchardt and R. Salomon, "Implementation of path planning using ga on mobile robots," in *IEEE Congress on Evolutionary Computation*, 2006, pp. 1831–1836.

[28] M. A. H. Antunes, "Uso de satélites para detecção de queimadas e para avaliação do risco de fogo," *Ação Ambiental*, vol. 12, pp. 24–27, 2000.

[29] A. Batista, "Detecção de incêndios florestais por satélite," *Revista Floresta (ISSN: 00153826)*, vol. 34, no. 2, pp. 237–241, 2004.

[30] A. H. Perera and T. K. Remmel, "Fire mapping in a northern boreal forest: assessing avhrr/ndvi methods of change detection," *Forest Ecology and Management*, vol. 152, pp. 119–129, 2001.

[31] LIF, "Laboratório de incêndios florestais: Pesquisas e projetos em prevenção e combate de incêndios florestais (ufpr)," Oct. 2006. [Online]. Available: www.floresta.ufpr.br/firelab

[32] L. P. Koproski, "O fogo e seus efeitos sobre a heperto e a mastofauna terrestre no parque nacional de ilha grande," Master's thesis, UFPR, 2005.

[33] G. Pessin, "Evolução de estratégias e controle inteligente em sistemas multi-robóticos robustos," Master's thesis, Universidade do Vale do Rio dos Sinos, Disponível em http://pessin.googlepages.com/disspessin2008.pdf, 2008.

[34] OSG, "Open scene graph," Feb. 2007. [Online]. Available: www.openscenegraph.com

[35] Demeter, "Demeter terrain engine," Mar. 2007. [Online]. Available: www.tbgsoftware.com

[36] SDL, "Simple directmedia layer," Feb. 2007. [Online]. Available: www.libsdl.org

[37] M. Wall, "Genetic algorithms library," May 2007. [Online]. Available: lancet.mit.edu/ga

[38] K. Dejong, "The analysis and behaviour of a class of genetic adaptive systems," Ph.D. dissertation, University of Michigan, USA, 1975.

[39] Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs*. New York, USA: Springer-Verlag, 1996.

[40] D. E. Goldberg, "Real-coded genetic algorithms, virtual alphabets, and blocking," *Complex Systems*, vol. 5, pp. 139–167, 1991.

[41] SNNS, "Stuttgart neural network simulator," May 2006. [Online]. Available: www-ra.informatik.uni-tuebingen.de/SNNS

[42] M. B. Dias, R. Zlot, N. Kalra, and A. Stentz, "Market-based multirobot coordination: a survey and analysis," *Proceedings of the IEEE*, vol. 94, pp. 1257–1270, 2006.