The 2009 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 11-15, 2009 St. Louis, USA

# A Fitness-Sharing based Genetic Algorithm for Collaborative Multi Robot Localization

Francesco Bori , Andrea Gasparri and Stefano Panzieri

*Abstract*— **In this paper, a novel genetic algorithm based on a "collaborative" fitness-sharing technique to deal with the Multi-Robot Localization problem is proposed. Indeed, the use of the fitness-sharing is twofold and competitive. It preserves the diversity among individuals during the space exploration process, thus maintaining evolutionary niches over time, and reinforces the best hypotheses by means of collaboration among robots, thus augmenting the selection pressure. Simulations by exploiting the robotics framework Player/Stage have been performed along with a proper statistical analysis for performance assessment.**

## I. INTRODUCTION

The localization problem consists of estimating the pose for a robot moving in an environment using data coming from sensors. Localization has been recognized as one of the most important problems in Robotics. In fact, the availability of reliable pose information turns out to be fundamental to perform almost any task. Moreover, the interaction of the robot with the environment and the noisy nature of sensor data make the problem highly complicated.

The emergence of Multi-Robot Systems (MRS) introduces new challenges for the localization problem. In fact, the inherent collaborative and cooperative nature of these systems requires new paradigms to be properly exploited. Indeed, frameworks for solving the localization problem in the multi-robot context might be naively obtained by extending classical approaches developed for the single robot context, e.g parallelizing their execution. However, this way the inherent collaborative nature of the system is completely neglected. Instead, better results can be obtained by taking into account all the available information.

In this paper, the map-based localization problem for a team of robots equipped with some exteroceptive sensors, e.g., laser scanners, is addressed. A novel approach based on a "collaborative" fitness-sharing technique is proposed. The key idea is to use a fitness-sharing technique for a twofold competitive objective. On one side it helps to preserve the diversity among individuals during the exploration of the search space, and thus it allows to maintain evolutionary niches over time. On the other side, it helps to reinforce the best hypotheses by means of collaboration among robots and therefore it allows to augment the selection pressure.

This works represents an extension of the idea proposed in [1]. The common baseline is to provide a mechanism for which evolutionary niches representing the most likely hypotheses (robot locations) are maintained over time. In previous works this was achieved by providing a spatial structure to the population and constraining the mating over this topology. In this work a niching method is exploited instead. This results in a more focused and effective action, while providing at the same time a suitable framework to strengthen the more promising hypotheses through collaboration.

The rest of the paper is organized as follows. In Section II an overview of the state of the art for the multi-robot localization problem is given. In Section III some theoretical insights about evolutionary computing are given. In Section IV the proposed "Collaborative" Fitness-sharing based genetic algorithm is described. In Section V simulation results are reported. Finally, in Section VI conclusions are drawn and future work is discussed.

## II. RELATED WORK

In [2] the concept of *mobile landmark* is introduced. The authors consider a team of robots exploring an unknown environment without any beacon. The exploration is carried out using the robots themselves as landmarks. Each vehicle repeats move-and-stop actions and acts as a landmark for the other robots, while a data fusion algorithm collects data to improve the estimate of the relative positioning of the robots. According to the authors, this mechanism works well in uncharted environments since the concept of landmark is intrinsically exploited. In [3], the idea previously introduced is exploited to improve the exploration of an unknown environment. In detail, underlining how the odometry errors might heavily affect the mapping of the environment, the authors introduce a mapping technique which acts also to minimize the effects of inherent navigation. A similar solution is proposed in [4], [5] where a new sensing strategy, named *robot tracker*, is exploited to improve the accuracy of the pose estimation of each robot. The robots explore the environment in teams of two; each platform is equipped with a robot tracker sensor that reports the relative position of the other robot. Measurements are used in a particle filter to update the poses of the multi-robot system together with the associated uncertainties. All the solutions above mentioned suffer from the following limitations: only one robot is allowed to move at any given time, and the team has to maintain sensorial contact at all times.

A different collaborative scheme, based on estimation theoretical framework, is presented in [6], where two robots are supposed to navigate in a partially known environment. At every meeting they stop and improve their localization

Authors are with the Department of Computer Science, University of "Roma Tre", Via Vasca Navale 79, Roma 00146, Italy {bori, gasparri, panzieri}@dia.uniroma3.it

by exchanging their *beliefs*, i.e., the posterior probability density over the state space conditioned to measurements. A particle filter is at the base of the algorithm, giving the possibility to handle a non Gaussian shaped belief, and achieve localization. Another promising solution is proposed in [7], [8] and reviewed in [9], [10], where a Kalman based algorithm is used to realize collaborative localization. During the navigation cycle, each robot collects data from its proprioceptive sensors to perform the prediction step of a Kalman filter while sharing information from the exteroceptive sensors with the rest of the team during the update. The authors introduce a distributed algorithm based on singular value decomposition of the covariance matrix. In this way, the centralized filter is decomposed into a number of smaller communicating filters, one for each robot. However, this approach can be applied only if inter-robot communication can be consistently guaranteed. If not, problems related to the maintenance of cross-correlations terms arise. In [11], a distributed approach based on maximum likelihood estimation is described. Robots are equipped with sensors that allow them to measure the relative pose and identity of nearby robots, as well as sensors that allow them to measure changes in their own pose. Therefore, localization is obtained using only the robots themselves as landmarks. In [12], the authors focus on the problem of examining the effect on localization accuracy of the number $N$ of participating robots and the accuracy of the sensors employed. In detail, the improvement in localization accuracy per additional robot as the size of the team increases is investigated.

## III. THEORETICAL BACKGROUND

### A. Genetic Algorithms

Genetic algorithms are a class of research techniques, inspired by Darwin's *Theory of Evolution*, applied in several research fields to solve optimization problems. These algorithms use a population of encoded strings (*chromosomes*) as candidate solutions to explore the search space. The candidate's evaluation is performed by means of an objective function (*fitness function*) and improvements at each iteration (*epoch*) result from the application of probabilistic transition operators (*crossover* and *mutation*) acting onto chromosomes. A simple genetic algorithm (SGA) usually provides three steps: initialization, selection and reproduction [13]. Initialization generates a population randomly picking up elements over the whole search space, selection draws an intermediate population relying on a fitness-based approach and reproduction causes the population to evolve combining elements from the intermediate population. Usually, crossover is performed with probability $p_c$, while *mutation* modifies chromosomes with probability $p_m$. This means that some individuals, likely with high fitness, will be exactly copied in the new population. The reader is referred to [14] for a complete overview of genetic algorithms.

### B. Genetic Algorithms Niching Methods and Fitness-Sharing

A simple genetic algorithm, when dealing with multimodal functions, would converge to the best peak, whereas, in addition to wanting to know the best solution, one may be interested in knowing the location of other optima. To overcome these limitations several techniques relying on the concept of niches have been introduced.

In multimodal GAs, a niche is commonly referred to as the location of each optimum in the search space, the fitness representing the resources of that niche. Niching methods have been developed to minimize the effect of genetic drift resulting from the selection operator in the traditional GA in order to allow the parallel investigation of many solutions in the population. An important number of niching methods have been reported in the literature, among them fitness-sharing, pre-selection and crowding [15].

In particular, the fitness-sharing technique modifies the search landscape by reducing the payoff in densely-populated regions. It derates each population element's fitness by an amount almost equal to the number of similar individuals in the population. Typically, the shared fitness $f_{sh,i}$ of an individual $i$ is defined as:

$$f_{sh,i} = \frac{f_i}{n_i}$$

where $f_i$ is the raw fitness and $n_i$ is the niche count given by:

$$n_i = \sum_{j=1}^{m} sh(d_{ij})$$

where $m$ denotes the population size, $d_{ij}$ represents the distance between the individual $i$ and individual $j$ and $sh$ describes the sharing function. This last term measures the similarity level between two elements of a population according to a threshold of dissimilarity $\sigma_s$ and is defined as follows:

$$sh(d_{ij}) = \begin{cases} 1 - \left(\frac{d_{ij}}{\sigma_s}\right)^{\alpha} & \text{if } d_{ij} < \sigma_s, \\ 0 & \text{otherwise} \end{cases}$$

where $\alpha$ is a constant parameter which regulates the shape of the sharing function (typically $\alpha = 1$). The effect of this scheme is to encourage search in unexplored regions. A complete overview of niching methods can be found in [16].

## IV. THE PROPOSED ALGORITHM

In the proposed framework, each robot runs an instance of the "Collaborative" Fitness-Sharing based Genetic Algorithm (CFS-GA). The key idea is to take advantage of a fitness-sharing technique for both maintaining evolutionary niches over time and augmenting the selection pressure of individuals. Indeed, as already pointed out in [1], being a niche a region in which a particular solution is preserved, a natural way to carry on multi-hypotheses is thus obtained. On the other hand, collaboration among robots is exploited in such a way that the selection pressure of individuals is augmented and therefore the survival of the best hypotheses is enhanced.

## A. Autonomous Localization

In the robotics context, a chromosome encodes the full state of the robot $p = (x, y, \theta)$, where $(x, y)$ represent the robot cartesian coordinates on a plane, while $\theta$ is its heading direction. In addition, the fitness function is defined as a pattern function giving a measure of the similarity between two vectors, as follows:

$$f(z_k, \hat{z}_k) = \frac{1}{L}\sum_{i=1}^{L}\frac{1}{\sqrt{2\pi}\sigma}e^{\frac{-(z_k^i-\hat{z}_k^i)^2}{2\sigma^2}}$$

where, $L$ is the number of laser beams, $z_k$ represents the sensor data, $\hat{z}_k$ is the expected one for the considered hypothesis and finally $\sigma$ is a measure of confidence related to the sensor data noise.

The proposed algorithm for autonomous localization works as follows. At each iteration $k$, a given robot $i$ performs two steps: kinematic update and population update.

The kinematic update is carried out by applying the current proprioceptive information, i.e., odometric information, to the kinematic model (the unicycle mode in the proposed implementation) for each individual of the population.

The population update is achieved by collecting data coming from exteroceptive sensors and then performing the evolutionary step. In order to achieve that, the raw fitness and the shared fitness must be computed. In particular, the raw fitness is used to identify the best individuals to be preserved (*elitism*) and the remaining individuals to be replaced (*epoch evolution*). Regarding the epoch evolution, an intermediate population is built by applying the tournament selection (with tournament size equals to 2) over the shared fitness [17]. New offspring are then obtained by applying the probabilistic transition operators crossover and mutation over this population. In the proposed implementation, crossover simply produces an offspring by combining the parents' chromosomes, and mutation produces an offspring by modifying some genes of a parent's chromosome. Finally, once the new population is built, the best individual describing the most likely robot pose is selected according to the raw fitness value multiplied by an aging factor (memory effect) which reduces the chattering phenomenon of the best individual selection over time (due to the sensitivity of the algorithm toward the noise affecting the measurements).

## B. Collaborative Localization

Collaboration among robots is available each time two or more robots are both in their range of communication ($c_r$) and in line of sight. Collaboration is achieved by exchanging relative distance and orientation coming from sensors affected by noise along with a portion of the populations for which some particular conditions are satisfied.

Let us assume two robots, respectively $r_1$ and $r_2$, to be in their range of communication and line of sight. Now, without any loss of generality let us consider the collaboration from the point of view of robot $r_1$ as the same holds for robot $r_2$ (in a similar way). At each iteration $k$, robot $r_1$ first collects data coming from the exteroceptive sensors in order to compute the fitness (both raw and shared) for its populations, successively it looks for neighboring robots to share data with. In this case robot $r_2$ is available, and then relative position and orientation coming from sensors affected by noise are exchanged along with a portion of the population for which the raw fitness value is greater than the average value of the whole population. This information will be exploited remotely by robot $r_1$ to augment the selection pressure and support the best hypotheses. In order to achieve that, a "virtual" population is built by robot $r_1$ first by collecting all the selected populations coming from the other robots together (in this case only data coming from robot $r_2$ is supposed to be available), and then by applying to them a roto-translation depending on the corresponding relative distance and orientation:

$$P_v^{(1)} = \bigcup_{i\in\mathcal{N}_1}\mathbf{R}(P_b^{(i)}, \Delta_{p,o}(r_1, r_i))$$

where $P_v^{(1)}$ denotes the "virtual" population of the robot $r_1$, $\mathcal{N}_1$ is the detected neighborhood for the robot $r_1$, $\mathbf{R}$ is the roto-translation operator, $P_b^{(i)}$ is the portion of population sent by the $i$-th neighbor and $\Delta_{p,o}(r_1, r_i)$ represent the relative position and orientation between the robots $r_1$ and $r_i$ . This "virtual" population describes the most likely areas where the local robot might be located from the other robots point of view. Indeed, this information can be exploited to strengthen local best hypotheses. This is done, by computing "virtual" niches $n_{v,i}$ around local hypotheses as follows:

$$n_{v,i}^{(1)} = \sum_{j=1}^{m_v} sh(d_{ij})$$

where $n_{v,i}^{(1)}$ is the "virtual" niche count around the $i$-th individual of robot $r_1$, $i$ is the index of the $i$-th local hypothesis, $j$ is the index of the $j$-th individual of the "virtual" population and $m_v$ is the size of the virtual population. As a result, the local hypothesis $i$ is strengthened as follows:

$$\tilde{f}_{sh,i} = f_{sh,i}\cdot n_{v,i}$$

Note that the search landscape is now affected in the opposite way, i.e., by augmenting the payoff in densely-populated regions. This increases each population element's fitness by an amount almost equal to the number of similar individuals in the "virtual" population. Indeed, this can be thought as a consensus-like approach where the information coming from other robots is taken as a "suggestion" in order to either give value to or diminish the confidence of local hypothesis. In the case such a suggestion is correct, this collaboration might significantly speed-up the localization process for the local robot. Conversely, if the local robot is already well-localized, a wrong suggestion would eventually bring ambiguity by strengthening misleading hypothesis for a few iterations, while if the local robot does not have any clue about its location, wrong information does not make it any worse.

### C. Complexity Analysis

In order to determine the computational complexity of the proposed CFS-GA running onboard a single robot with a population of $m$ individuals, the following main functions are analyzed:

*1) Fitness:* The computation of the raw fitness function is achieved by computing the difference between the real robot measurements and the measurements estimated by each individual. Assuming the number of beams to be $L$, the overall complexity is $O(m \cdot L)$

*2) Shared Fitness:* The evaluation of the shared fitness requires to calculate the distance among all the individuals of the population, to compute a niche count for each individual and to perform a division between the raw fitness and the related niche count. The dominant operation is the computation of the distance among the individuals and therefore the complexity is $O(m^2)$.

*3) Data Sharing:* The data-sharing operation involves the exchange of both relative distance and orientation along with the portion of the population for which the raw fitness value is greater than the average value over the whole population. The dominant operation is the comparison operation for which the complexity is $O(m)$.

*4) Shared Fitness Update:* The update of the shared-fitness involves the computation of the distance between the $m$ individuals of the local population (regarding the robot in analysis) and the $m_v$ individuals of the virtual population (obtained by putting together the data collected from the neighboring robots). The dominant operation is again the distance and, in this case, the related complexity is $O(m \cdot m_v)$.

*5) Selection:* The selection process is implemented by exploiting the "Tournament Selection" with tournament size equals to 2, and its complexity is $O(m)$.

*6) Crossover and Mutation:* Both the crossover and mutation operators have a constant complexity when applied to a single individual, therefore for the whole population the complexity is $O(m)$ each.

As a result, putting together all the single pieces, the overall computational complexity of the algorithm running onboard each single robot turns out to be $\max\{O(m^2), O(m \cdot m_v)\}$.

## V. SIMULATION RESULTS

The proposed algorithm has been throughly investigated by exploiting the robotics simulation framework Player/Stage [18]. It consists of a set of tools for multi-robot and distributed sensor systems. Briefly speaking, Player provides a network interface to a variety of robot and sensor hardware. Player's client/server model allows robot control programs to be written in any programming language and to run on any computer with a network connection to the robot. Player supports multiple concurrent client connections to devices, creating new possibilities for distributed and collaborative sensing and control. On the other side, Stage simulates a population of mobile robots moving in and sensing a two-dimensional bitmapped environment. Various sensor models are provided, including sonar, scanning laser rangefinder, pan-tilt-zoom camera with color blob detection and odometry. Stage devices present a standard Player interface so few or no changes are required to move between simulation and hardware.

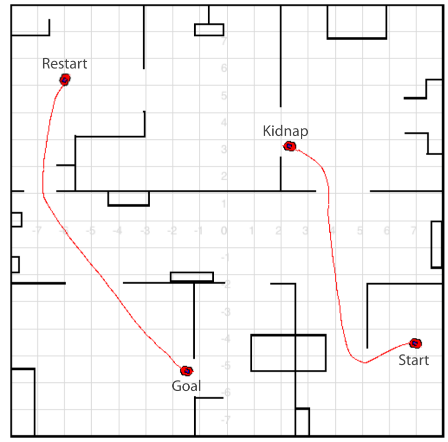



Fig. 1. First scenario. Autonomous Localization with Kidnap. Robot's path from start point (S) to kidnap point (K) and from restart point (R) to goal (G).

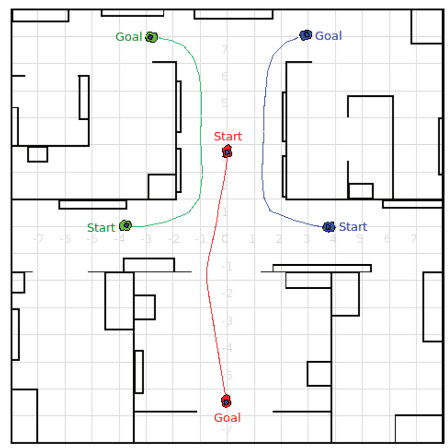



Fig. 2. Second scenario. Collaborative Localization. Robots' path from start point (S) to goal (G). Communication constrained by range of visibility and line of sight.

Two different scenarios were considered for performance assessment. In the first scenario, the autonomous localization along with the kidnapped robot problem [19] was investigated. In particular the proposed algorithm has been compared against the "Adaptive Monte Carlo Localization algorithm" (AMCL) [20], already available in the Player/Stage framework, and against the "Spatially Structures Genetic algorithm" (SSGA) proposed in [1]. In the second scenario, the advantages introduced by the collaborative strategy were investigated. The environment shown in Fig. 1 was exploited for the first case, while the environment depicted in Fig. 2 was used for the second case. Both scenarios represent a typical indoor, office-like environment.

A set of 100 independent runs was executed for each scenario, and average values were computed. Specifically, at

TABLE I

SIMULATION SETTING

| Parameter | Description | SFGA |
|---|---|---|
| $m$ | Population Size | 300 |
| $L$ | No. of Pattern Beams | 18 |
| $l$ | Beam Range [m] | 8 |
| $\sigma$ | Confidence Measure [m] | 0.5 |
| $p_b$ | Best Individuals Percentage [%] | 80 |
| $p_s$ | Selected Individuals Percentage [%] | 20 |
| $p_r$ | Random Individuals Percentage [%] | 5 |
| $T$ | Tournament Size | 2 |
| $p_c$ | Crossover Probability [%] | 80 |
| $p_m$ | Mutation Probability [%] | 10 |
| $\sigma_s$ | Dissimilarity Threshold [% size(Env)] | 5 |
| $\alpha$ | Shape Parameter | 1 |
| $c_r$ | Communication Range [m] | 4 |
| $Q_m$ | Odometry Noise Var. [m/s, $rad$/s] | [0.05, 0.1] |
| $Q_l$ | Laser Noise Var. [m] | 0.1 |
| $Q_{rd}$ | Rel. Distance Noise Var. [m] | 0.2 |
| $Q_{ro}$ | Rel. Orientation Noise Var. [$rad$] | 0.1 |

each iteration of a given trial, a pose error was computed (using the Euclidian metric) with respect to the best hypothesis. Note that, the initial population was always drawn from a random uniform distribution of individuals over the whole environment. Regarding the disturbances affecting both the proprioceptive and exteroceptive sensors, gaussian noises with zero means and covariances $Q_m$, $Q_l$, $Q_{rd}$, $Q_{ro}$ respectively for the odometric measurements, for the laser scanner measurements, and for the relative distance and orientation measurements have been considered. Table I describes the parameters setting adopted for simulations.

Fig. 3 shows the localization error averaged over 100 trials for the first scenario. In detail, the solid (gren) line describes the localization error for the proposed CFS-GA, the dotted (red) line represents the localization error for the SSGA and, the dashed (blue) line is the localization error for the AMCL. According to the obtained results, the three algorithms perform similarly in terms of accuracy until the kidnap happens. In particular, it can be noticed from the subplot in the nested box (a), that the AMCL converges more quickly to the correct robot location, while the proposed CFS-GA takes a little bit longer and the SSGA even longer. This can be explained by the tendency of the last two approaches to maintain several hypotheses over time (for global localization purposes) which leads to a longer time before to *trust* the correct hypothesis. Nevertheless, the CFS-GA outperforms the SSGA proving to be a more focused and effective localization strategy.

On the other hand, this capability to maintain several hypothesis over time turns out to be crucial when the kidnap happens. In fact, the CFS-GA always detect the kidnap event and properly recovers the robot location due to the tendency to continuously explore new locations, even when the correct robot location is being tracked. Conversely, the AMCL, which simply adds a number of randomly placed samples at every time instant as detailed in [21], often fails to re-locate the robot. A similar consideration holds for the SSGA weakened by the requirement of an additional kidnap sensing strategy which might fail to recognize the kidnap event. In particular, the plot in the nested box (b) details the algorithms behavior after the kidnap.

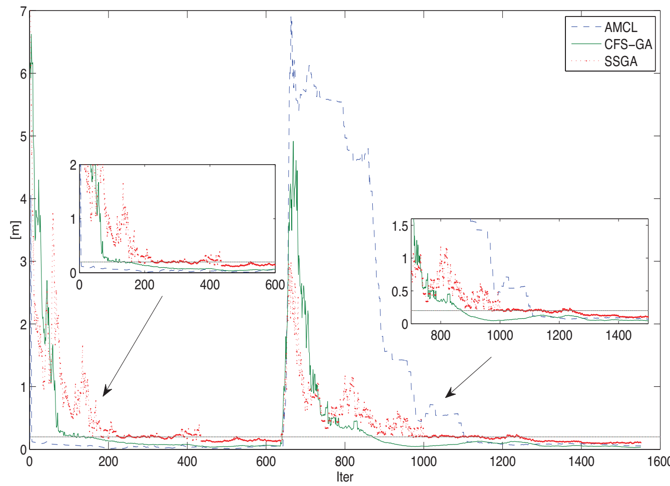
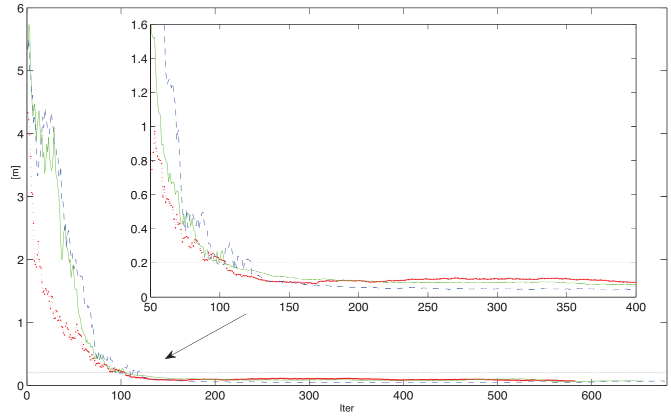


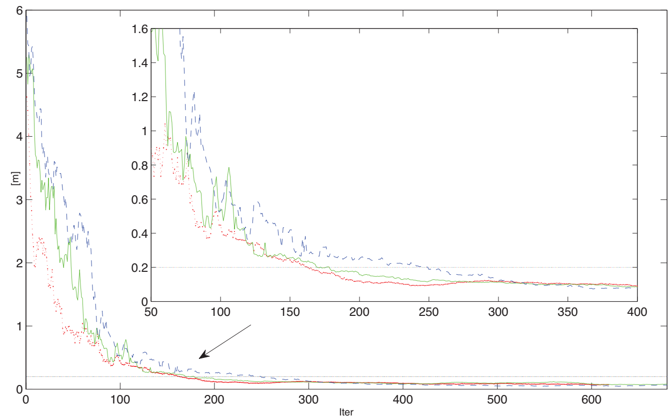

Fig. 3. First scenario. Autonomous Localization with Kidnap. Solid (green) line: CFS-GA Localization Error. Dashed (blue) line: AMCL Localization Error. Dotted (red) line: SSGA Localization Error.

Fig. 4 shows the localization error averaged over 100 trials for the second scenario. In particular, Fig. 4-(a) and Fig. 4-(b) show respectively the results obtained for the proposed CFS-GA with or without collaboration among robots. According to the results obtained for the previous scenario, the autonomous localization already performs satisfactorily on its own. For this reason, robots have been purposely placed in the middle of three different corridors where laser data are temporarily partially useless to make the localization problem particularly difficult. Indeed, this would help to better highlight the contribution coming from collaboration. Note that, the collaboration cannot improve the accuracy of estimation, i.e., the average localization error after the convergence of the algorithm is roughly the same in both cases. Nonetheless, a significant speed up of the algorithm convergence can be noticed. Indeed, while the autonomous localization requires almost 300 iterations for all robots in order to settle around a value of $15cm$, the same is obtained by the collaborative localization after only roughly 100 iterations. This can be explained by the fact that, any time two robots meet, the way in which they cooperate allows them to strengthen the more likely hypotheses by computing the virtual niches which affect the landscape by augmenting the pay-off in densely populated areas.

Finally, in order to investigate the robustness of the proposed CFS-GA, 500 simulations involving random starting point and goal points for both environments were considered. According to the obtained results, the CFS-GA proved to always be able to localize the robot with a population of 300 individuals, while the AMCL showed a percentage of failure of roughly $35\%$ starting with 10000 particles (adaptive population ranging from 10000 to 1000 particles) and the SSGA a percentage of failure of roughly $15\%$ starting with a population of 300 individuals.




(a) Collaborative Localization




(b) Autonomous Localization

Fig. 4. Second scenario. Collaborative localization against autonomous localization. Plot lines' colors match robots' paths color.

## VI. CONCLUSIONS AND FUTURE WORK

In this paper, a novel genetic algorithm based on a "Collaborative" Fitness-Sharing technique to deal with the Multi-Robot Localization problem has been proposed.

The key idea is to use a fitness-sharing technique for a twofold competitive objective. On one side this is used to preserve the diversity among individuals during the exploration of the search space, and thus it allows to maintain evolutionary niches over time. On the other side, this is exploited to reinforce the best hypotheses by means of collaboration among robots and therefore it allows to augment the selection pressure.

This works represents an extension of the idea proposed in [1]. The common baseline is to provide a mechanism for which evolutionary niches representing the most likely hypotheses (robot locations) are maintained over time. In previous works this was achieved by providing a spatial structure to the population and constraining the mating over this topology. In this work a niching method has been exploited. This results in a more focused and effective action, while providing at the same time a suitable framework to strengthen the more promising hypotheses through collaboration.

Several simulations by exploiting the robotics simulation framework Player/Stage have been performed for performance assessment. According to the simulation results, the proposed CFS-GA seems to be a promising technique for both autonomous localization and collaborative multi-robot localization.

Interesting challenges still remain for future work. First, a real implementation in order to investigate the effectiveness of the proposed CFS-GA in a real context is currently under study. In addition, an investigation to bring this idea into a probabilistic context will be investigated. This way a major shortcoming of this approach, i.e., inability of providing a measure of uncertainty of the estimation, would be overcome.

## REFERENCES


[1] A. Gasparri, S. Panzieri, F. Pascucci, and G. Ulivi, "A spatially structured genetic algorithm over complex networks for mobile robot localization," in *IEEE ICRA*, Rome, Italy, 2007.

[2] R. Kurazume, S. Nagata, and S. Hirose, "Cooperative positioning with multiple robots," in *IEEE ICRA*, vol. 2, 1994, pp. 1250–1257.

[3] I. Rekleitis, G. Dudeck, and E. Milios, "Multi-robot exploration of an unknown environment, efficiently reducing the odometry error," in *IJCAI 1997*, 1997, pp. 1340–1345.

[4] I. M. Rekleitis, G. Dudek, and E. E. Milios, "Multi-robot cooperative localization: a study of trade-offs between efficiency and accuracy," in *IEEE IROS*, 2002, pp. 2690–2696.

[5] ——, "Probabilistic cooperative localization and mapping in practice," in *IEEE ICRA*, September 14–19 2003, pp. 1907–1912.

[6] D. Fox, W. Burgard, H. Kruppa, and S. Thrun, "A probabilistic approach to collaborative multi-robot localization," *Autonomous Robots*, vol. 8, no. 3, pp. 325–344, 2000.

[7] S. Roumeliotis and G. Bekey, "Distributed multi-robot localization," *IEEE Transactions on Robotics and Autonomation*, vol. 18, no. 5, pp. 781–795, 2002.

[8] A. Mourikis and S. Roumeliotis, "Performance analysis of multirobot cooperative localization," *IEEE Trans. on Robotics*, vol. 22, no. 4, pp. 666–681, August 2006.

[9] A. Martinelli, F. Pont, and R. Siegwart, "Multi-robot localization using relative observation," in *IEEE ICRA*, April 18–22 2005.

[10] A. Martinelli and R. Siegwart, "Observability analysis for mobile robot localization," in *IEEE IROS*, 2005, pp. 1264–1269.

[11] A. Howard, M. J. Mataric, and G. S. Sukhatme, *Experimental Robotics VIII*. Springer Berlin / Heidelberg, 2003, ch. Localization for Mobile Robot Teams: A Distributed MLE Approach, pp. 146–155.

[12] S. I. Roumeliotis and I. M. Rekleitis, "Propagation of uncertainty in cooperative multirobot localization: Analysis and experimental results," *Auton. Robots*, vol. 17, no. 1, pp. 41–54, 2004.

[13] D. E. Goldberg, *Genetic algorithms in search, optimization, and machine learning*. Addison-Wesley, 1989.

[14] M. Mitchell, *An Introduction to Genetic Algorithms*. Cambridge, MA, USA: MIT Press, 1998.

[15] L. Gao and Y. Hu, "Multi-target matching based on niching genetic algorithm," *International Journal of Computer Science and Network Security*, vol. 6, no. 7, pp. 215 – 220, 2006.

[16] S. W. Mahfoud, "Niching methods for genetic algorithms," Ph.D. dissertation, University of Illinois at Urbana-Champaign, Champaign, IL, USA, 1995.

[17] B. L. Miller, B. L. Miller, D. E. Goldberg, and D. E. Goldberg, "Genetic algorithms, tournament selection, and the effects of noise," *Complex Systems*, vol. 9, pp. 193–212, 1995.

[18] B. P. Gerkey, R. T. Vaughan, and A. Howard, "The player/stage project: Tools for multi-robot and distributed sensor systems," in *ICAR 2003*, 2003, pp. 317–323.

[19] S. Engelson and D. McDermott, "Error correction in mobile robot map learning," in *IEEE ICRA*, Nice, France, May 1992, pp. 2555 – 2560.

[20] D. Fox, "Adapting the sample size in particle filters through kld-sampling," *I. J. Robotic Res.*, vol. 22, no. 12, pp. 985–1004, 2003.

[21] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press, 2005.