Noname manuscript No.
(will be inserted by the editor)

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

Received: date / Accepted: date

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.

Keywords Multi-Robot · Localization · Genetic Algorithm

1 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.

Multi-Robot Systems can be classified in regards to their architecture into two categories: *centralized* and *decentralized* [Cao et al(1997)]. Centralized architectures are characterized by a single control robot (leader) that is in charge of organizing the

Address(es) of author(s) should be given

activities of the other robots. The leader takes part in the decision process for the whole team, while the other members act according to the dispositions of the leader. Conversely, decentralized architectures are characterized by "self-organization", i.e., each robot is autonomous in the decision process with respect to each other. However, all robots share a common goal and their actions are toward its achievement.

Localization techniques have been developed with respect to these two architectures. In a centralized system, a leader collects data provided by the team and performs the localization process for the whole group. In a decentralized system, each robot performs its estimation and exchanges data with the other robots to improve the localization process. Both paradigms present advantages as well as drawbacks. Normally, the assignment of a task is easier in a centralized system compared to a distributed one, as the leader is the only one in charge of it. Additionally, centralizing the computation requires only one robot, or few if redundancy is taken into account, with suitable hardware capabilities. However, this leads to a lack of robustness as, once a leader fails, the system becomes unable to accomplish the task. These disadvantages can be overcome removing the central processing unit and spreading all the decisional issues over the team. This way, since each robot acts independently, modularity and robustness are achieved [Parker(2000)]. Obviously, suitable hardware capabilities for all robots are required in this case.

Moreover, in a centralized fashion, a supervisor collects all the data coming from the robots and provides an estimate of the poses for the robots in the team. This approach requires all members to continuously communicate with the supervisor. In order to maintain the communication, robots need either to move closely to the supervisor or to implement a mobile ad-hoc network. Therefore some constraints on robots mobility have to be defined to guarantee at least one communication-path from any robot to the supervisor at each time instant. The decentralized approach instead, referred in literature as *collaborative or cooperative* localization, assumes that each robot in the workspace uses its own sensors, exchanges data only with other robots within its neighborhood, and runs a local algorithm to estimate its own pose.

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 [Gasparri et al(2007)], [Gasparri et al(2009)]. 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 2 an overview of the state of the art for the multi-robot localization problem is given. In Section 3 some theoretical insights about evolutionary computing are given. In Section 4 the proposed "Collaborative" Fitness-sharing based genetic algorithm is described. In Section 5 simulation results are reported. Finally, in Section 6 conclusions are drawn and future work is discussed.

1.1 Basic Assumptions

In the rest of the paper, the following assumptions will be taken into account for the multi-robot system:

- Robots are assumed to have a map of the environment in which they are moving.
- Robots are assumed to be equipped with a laser range-finder and a compass.
- Robots are assumed to be able to compute both relative distance and orientation.
- Robots are assumed to be able to detect and identify each other.

Note that, the capability to compute the relative distance and orientation among robots along with the capability to detect and identify each other, are quite common assumptions which can be found in the majority of the works available in the literature, among the others [Roumeliotis et al(2009)], [Howard (2006)], [Martinelli et al(2005)], [Fox et al(2000)]. In particular, the data association problem related to the last assumption can be eliminated by properly engineering the team of robots, e.g., equipping robots with markers. For example, if the robots are using laser range-finders, mutual detection can be facilitated using retro-reflective targets [Howard et al(2004)], and the subset of laser rays striking other robots discarded.

2 Related Work

In [Kurazume et al(1994)] 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 [Rekleitis et al(1997)], 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 [Rekleitis et al(2002),Rekleitis et al(2003)] 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 [Fox et al(2000)], where two robots are supposed to navigate in a partially known environment. At every meeting they stop and improve their localization 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 [Roumeliotis et al(2002),Mourikis et al(2006)] and reviewed in [Martinelli et al(2005),Martinelli et al(2005)], 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 [Howard et al(2003)], 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 [Roumeliotis et al(2004)], 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. In [Roumeliotis et al(2009)], a distributed Maximum A Posteriori (MAP) estimator for multi-robot Cooperative Localization (CL) is introduced. Robots are supposed to be able to uniquely identify other robots in the team and measure their relative distance and bearing. The proposed algorithm effectively exploit the computational and storage resources of all robots in the team to reduce the processing requirements and achieve real-time performance. However, a synchronous communication among robots must be guaranteed in order to distribute the computation of the MAP. Indeed, as stated by the authors themselves, this might be hard to achieve in the case of environments with frequent communication failures.

3 THEORETICAL BACKGROUND

3.1 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 [Goldberg(1989)]. 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 [Mitchell(1998)] for a complete overview of genetic algorithms.

3.2 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 [Gao et al(2006)].

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:

$$f_{sh,i} = \frac{f_i}{n_i} \tag{1}$$

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

$$n_i = \sum_{j=1}^{m} sh(d_{ij}) \tag{2}$$

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 σ_s and is defined as follows:

$$sh(d_{ij}) = \begin{cases} 1 - \left(\frac{d_{ij}}{\sigma_s}\right)^{\alpha} & \text{if } d_{ij} < \sigma_s, \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

where α 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 [Mahfoud(1995)].

4 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

[Gasparri et al(2007)], 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.

Algorithm 1: Genetic Algorithm with shared fitness for robotic localization

Data: Population $\mathcal{P}_t$ of size m and respectively fitness F_t at time t; odometric information u_t at time t.
Result: New population of size m $\mathcal{P}_{t+1}$ and new fitness F_{t+1} at time $(t+1)$

```
/* Kinematic Evolution                                              */
P̃_{t+1} = f(P_t, u_t);
/* Shared Fitness Computation                                       */
SF = sharedFitness(P̃_{t+1}, σ_s, α);
/* Collaborative Localization (if any)                              */
if Neigh(r_i) ≠ ∅ then
    /* Data Sharing                                                 */
    foreach r_j ∈ Neigh(r_i) do
        P_v ← Data(r_j) ;                          // Virtual Population
    end
    /* Hypotheses Reinforcement                                     */
    SF = updateSharedFit(P̃_{t+1}, P_v, SF);
end
/* Best Individuals                                                 */
P_b = bestSel(P̃_{t+1}, F, p_b);
/* Random Population                                                */
P_r = randPop(p_r);
/* Tournament Selection                                             */
P_p = selection(P̃_{t+1}, SF, p_s);
/* Crossover                                                        */
P_c = crossover(P_p, p_c);
/* Mutation                                                         */
P_g = mutation(P_c, p_m) ;
/* New Population                                                   */
P_{t+1} = {P_b ∪ P_r ∪ P_g};
/* Update Fitness                                                   */
F_{t+1} = fitness(P_{t+1});
```

4.1 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 θ 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:

$$f(z_k, \hat{z}_k) = \frac{1}{L}\sum_{i=1}^{L}\frac{1}{\sqrt{2\pi}\sigma}e^{\frac{-(z_k^i - \hat{z}_k^i)^2}{2\sigma^2}} \tag{4}$$

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 σ 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: kinematics update and population update.

The kinematics update is carried out by applying the current proprioceptive information, i.e., odometric information, to the kinematics 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 [Miller et al(1995)]. 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).

4.2 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. Note that, the assumption of being in line of sight is strictly related to the particular sensor equipment given in Subsection 1.1. Generally speaking , the proposed collaborative technique requires a couple of robots to be able to compute both relative distance and orientation among each other. Therefore, any sensing modality, for instance RF along with compass, which is able to provide this kind of information would be enough. In the proposed simulation for sake of simplicity we have considered the line-of-sight which allows a couple of robots with a laser range-finder to easily compute the inter-distance among each other, while the relative orientation is worked out by means of compass data.

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:

$$P_v^{(1)} = \bigcup_{i \in \mathcal{N}_1} \mathbf{R}(P_b^{(i)}, \Delta_{p,o}(r_1, r_i)) \tag{5}$$

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:

$$n_{v,i}^{(1)} = \sum_{j=1}^{m_v} sh(d_{ij}) \tag{6}$$

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:

$$\tilde{f}_{sh,i} = f_{sh,i} \cdot n_{v,i} \tag{7}$$

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.

4.3 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: Fitness, Shared Fitness, Data Sharing, Shared Fitness Update, Selection, Crossover and Mutation.

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 $O(m \cdot L)$ 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 $O(m^2)$. 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 values is greater than the average value over the whole population. The dominant operation is the comparison operation for which the complexity is $O(m)$. 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 $O(m \cdot m_v)$. The selection process is implemented by exploiting the "Tournament Selection" with tournament size equals to 2, and its complexity is $O(m)$. 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 $\max\{O(m^2), O(m \cdot m_v)\}$. In particular, it should be noticed how the size of the neighborhood of a robot affects the overall computational complexity of the algorithm. In fact, as it can been seen in eq. (5), for each robot the size of the virtual population grows proportionally with the number of its neighbors and so does the complexity of the virtual niches computation. Indeed, this represents a bottleneck for the use of the proposed collaborative localization technique.

5 SIMULATION RESULTS

The proposed algorithm has been throughly investigated by exploiting the robotics simulation framework Player/Stage [Gerkey et al(2003)]. 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.

Three different scenarios were considered for performance assessment. In the first scenario, the capability of maintaining multi-hypothesis over time was investigated. In the second scenario, the autonomous localization along with the kidnapped robot

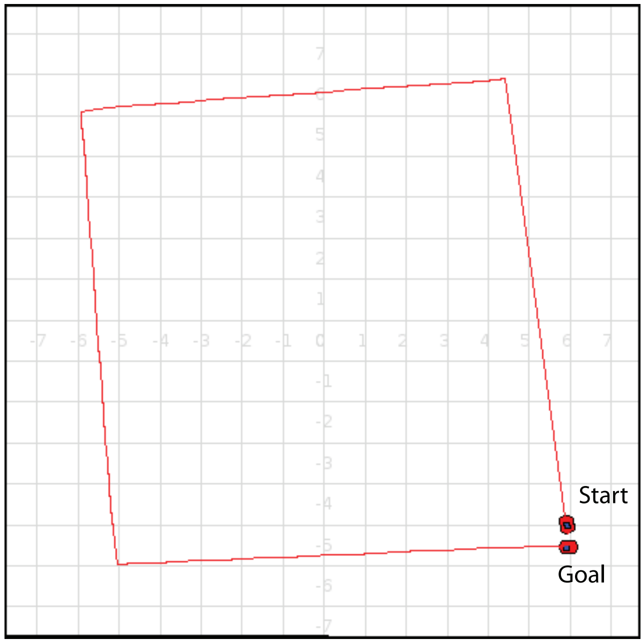


Fig. 1 First scenario. Robot's path from the start point (S) to the goal (G).

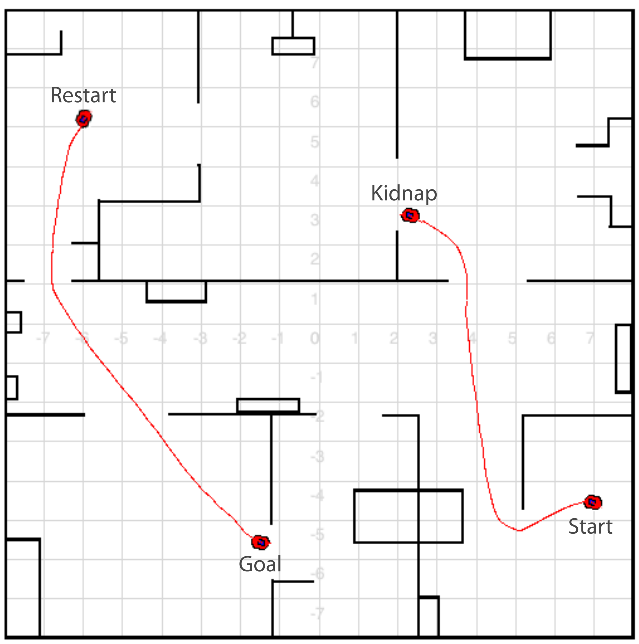


Fig. 2 Second scenario. Autonomous Localization with Kidnap. Robot's path from start point (S) to kidnap point (K) and from restart point (R) to goal (G).

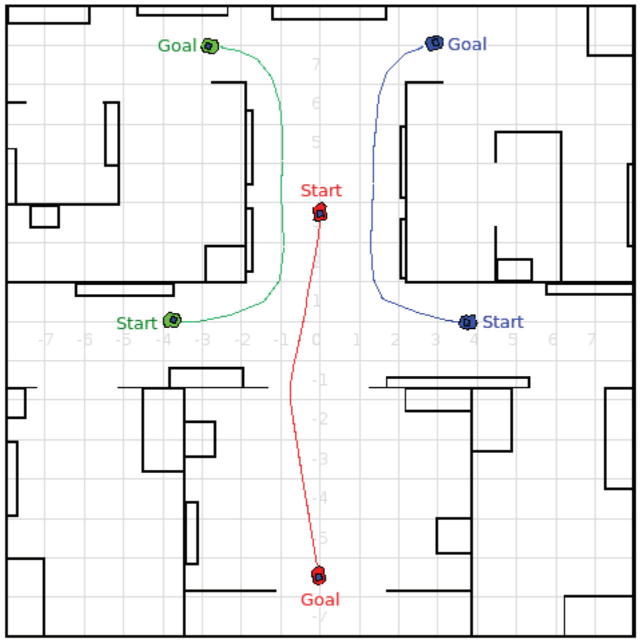


Fig. 3 Third scenario. Collaborative Localization. Robots' path from start point (S) to goal (G). Communication constrained by range of visibility and line of sight.

problem [Engelson et al(1992)] was investigated. In the third scenario, the advantages introduced by the collaborative strategy were investigated. The environment shown in Fig. 1 was exploited for the first analysis, while Fig. 2 describes the environment exploited for the second scenario and finally Fig. 3 depicts the environment that was used for the third case. All those scenarios represent a typical indoor, office-like environment. In particular, the proposed CFS-GA has been compared against:

- the "Adaptive Monte Carlo Localization algorithm" (AMCL) proposed in [Fox(2003)] (already available in the Player/Stage framework),
- the "Differential Evolution" algorithm (DE) proposed in [Ghidary et al(2007)],
- the "Particle Swarm Optimization" algorithm (PSO) proposed in [Ghidary et al(2007)],
- the "Spatially Structures Genetic algorithm" (SSGA) proposed in [Gasparri et al(2007)].

A set of 100 independent runs was executed for each scenario, and average values were computed. Specifically, at 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 noise affecting both the proprioceptive and exteroceptive simulated sensors measurements, 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 1 describes the parameters setting adopted for simulations.

Table 1 Simulation Setting

Parameter	Description	SFGA
m	Population Size	300
L	No. of Pattern Beams	18
l	Beam Range [m]	8
σ	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
σ_s	Dissimilarity Threshold [% size(Env)]	5
α	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

5.1 First Scenario

Fig. 4 shows an example of localization behavior of the proposed algorithm in a squared, fully symmetric room using only laser range-finders data where each red segment represents a single individual belonging to the population. Note that, only the first half of the entire path is shown because the behavior of the algorithm does not change significantly. In such a scenario, the uncertainty associated with the robot location is remarkably high because of the symmetry of the environment. In particular, when a robot is near to a perimetrical wall, the actual pose and other three poses nearby the remaining walls have roughly the same fitness value. Nevertheless, CFS-GA showed its ability to face this problem keeping track of the most likely hypotheses (grouped within niches) over time, e.g., Fig 4-b), Fig 4-c). Furthermore, when a robot performs a rotation, the uncertainty, due to the noise associated with the odometry, causes the algorithm to spread the population on a larger portion of the map. However, this does not significantly affect the convergence of the algorithm and the multi-hypothesis benefits are clearly visible, e.g., Fig 4-a), Fig 4-d). Note that, the problem of selecting the correct hypothesis among equiprobable individuals belonging to different niches could be overcome with the employment of the compass.

5.2 Second Scenario

In this scenario, the autonomous localization along with the capability to detect a kidnap have been investigated.

Fig. 5 shows the localization error averaged over 100 trials for the comparison against the AMCL, where the CFS-GA was run with a population of 300 individuals and the AMCL was initialized with 10000 particles (adaptive population ranging from 10000 to 1000 particles). In, detail the solid(blue) line describes the localization error for the proposed CFS-GA, while the dashed (red) line is the localization error for the AMCL. According to the obtained results, the two algorithms perform similarly

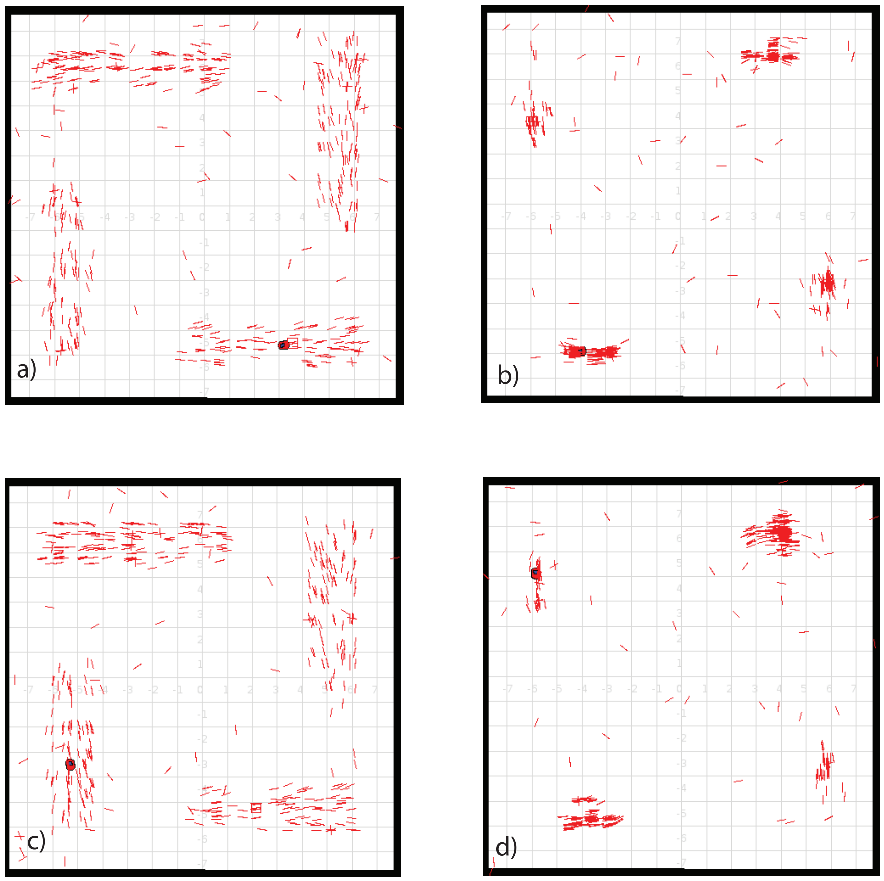


Fig. 4 First scenario. Symmetric squared environment. Red segments are the individuals belonging to the population.

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. This can be explained by the tendency of the CFS-GA to maintain several hypotheses over time (for global localization purposes) which leads to a longer time before to *trust* the correct hypothesis. 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 [Thrun et al(2005)], often fails to re-locate the robot.

Fig. 6 shows the localization error averaged over 100 trials for the comparison against the DE, where both the CFS-GA and the DE were run with a population of 300 individuals. In, detail the solid(blue) line describes the localization error for the proposed CFS-GA, while the dashed (red) line is the localization error for the DE.

According to the obtained results, the proposed CFS-GA always outperforms the DE algorithm, which has proven to suffer from a high fluctuation of the best solution. We believe this might be related to the random component of the new individuals generation process which involves the best existing individual and two random ones [Ghidary et al(2007)]. On the other side, we also believe that this random component explains the capability of the DE algorithm to recover from the kidnap problem, though a very slow converge time can be noticed.

Fig. 7 shows the localization error averaged over 100 trials for the comparison against the PSO, where the CFS-GA was run with a population of 300 individuals and the PSO with a population of 3000 particles . In, detail the solid(blue) line describes the localization error for the proposed CFS-GA, while the dashed (red) line is the localization error for the PSO. According to the obtained results, the PSO algorithm turned out to be very inadequate to deal with the global localization problem: the localization error, as shown in the nested box, was always above the threshold fixed at 20 cm, and the algorithm has proven to fail to detect and recover from the kidnap problem. We believe this might be due to the incapability of the PSO algorithm to carry on the multi-hypothesis over time. Indeed, this aspect turns out to be crucial anytime a robot moves within an environment with structural symmetries, as shown in Fig. 1.

Fig. 8 shows the localization error averaged over 100 trials for the comparison against the SSGA, where both the CFS-GA and the SSGA were run with a population of 300 individuals. In, detail the solid(blue) line describes the localization error for the proposed CFS-GA, while the dashed (red) line is the localization error for the SSGA. According to the obtained results, the two algorithms show similar performances. However, the SSGA is weakened by the requirement of an additional kidnap sensing strategy which might fail to recognize the kidnap event. Furthermore, the CF-SGA provides a more accurate localization compared to the SSGA proving to be a more focused and effective localization strategy.

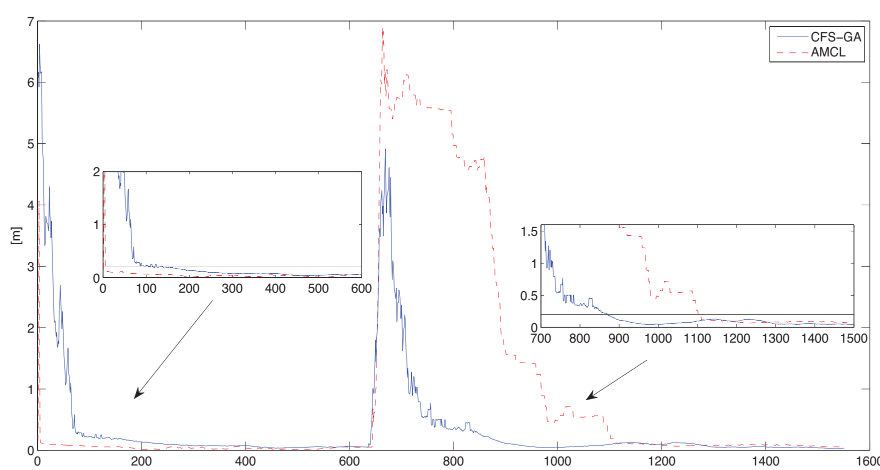


Fig. 5 First scenario. Autonomous Localization with Kidnap. Solid (blue) line: CFS-GA Localization Error. Dashed (red) line: AMCL Localization Error.

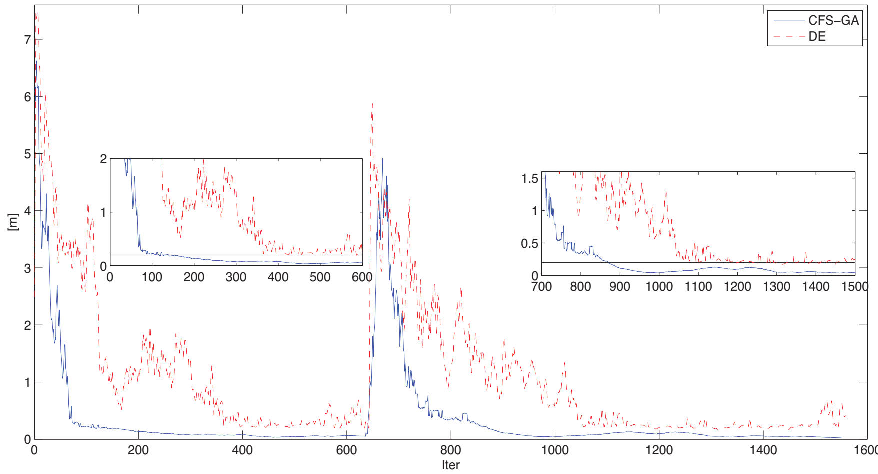


Fig. 6 First scenario. Autonomous Localization with Kidnap. Solid (blue) line: CFS-GA Localization Error. Dashed (red) line: DE Localization Error.

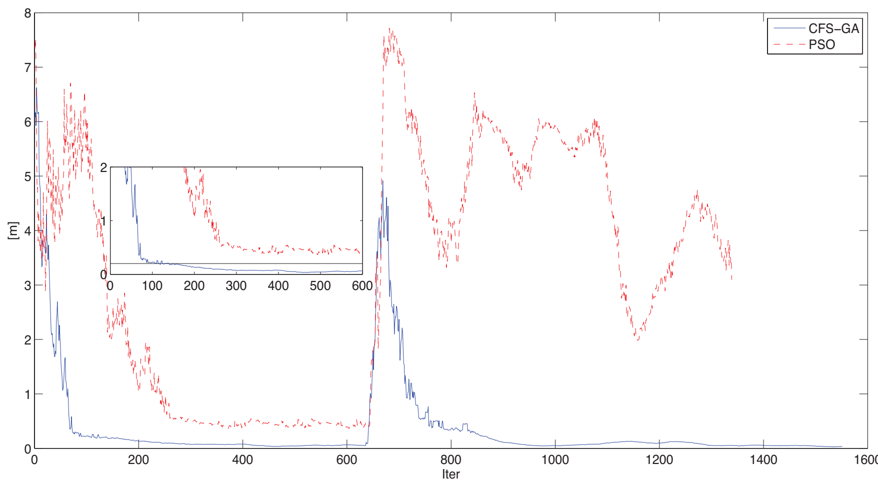


Fig. 7 First scenario. Autonomous Localization with Kidnap. Solid (blue) line: CFS-GA Localization Error. Dashed (red) line: PSO Localization Error.

5.3 Third Scenario

Fig. 9 shows the localization error averaged over 100 trials for the second scenario. In particular, Fig. 9-(a) and Fig. 9-(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. Obviously, the collaboration cannot

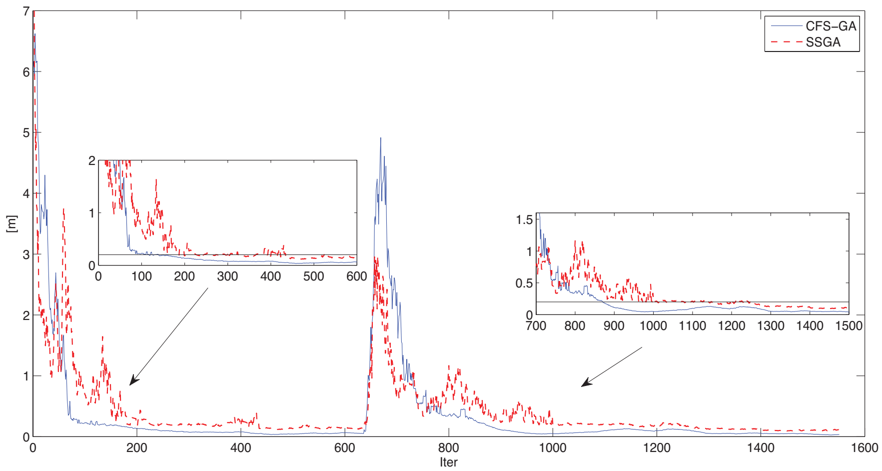
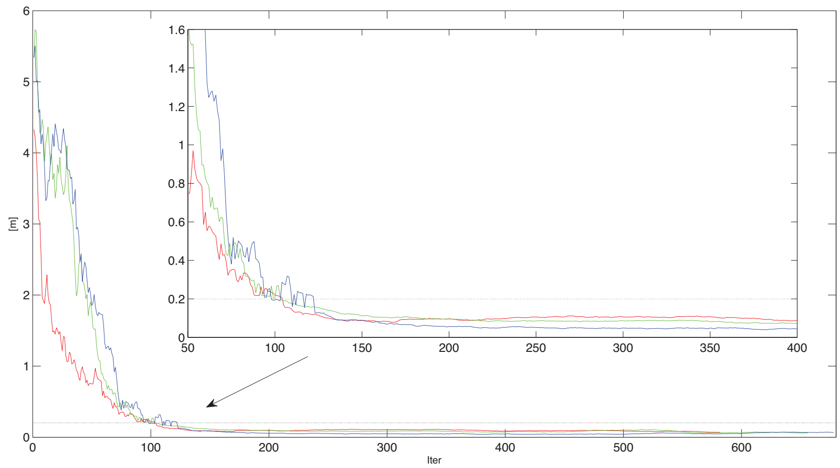


Fig. 8 First scenario. Autonomous Localization with Kidnap. Solid (blue) line: CFS-GA Localization Error. Dashed (red) line: SSGA Localization Error.

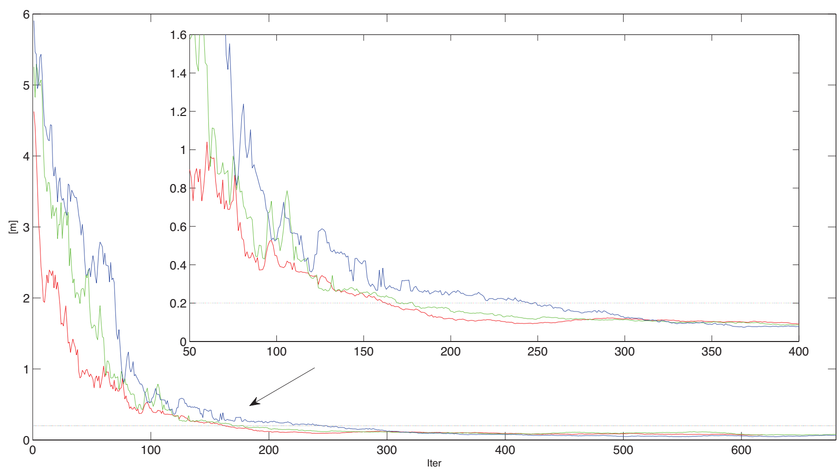
improve the accuracy of estimation, i.e., the average localization error after the convergence of the algorithm is roughly the same in both cases. Nonetheless, a significant speed up of the algorithm convergence can be noticed. Indeed, while the autonomous localization requires almost 300 iterations 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. However, it should be pointed out that according to the simulation results, the speed-up on the convergence time becomes less and less significant by increasing the size of the neighborhood.

5.4 Parameters Tuning

It is well-known that a critical point for a genetic algorithm is the tuning of the parameters [Goldberg(1989)]. Indeed, this can heavily affects the effectiveness of the algorithm itself [Nannen et al(2008)]. For this reasons several approach to provide self-tuning capabilities have been investigated [Yuan et al(2005)], [Perez et al(2008)]. As far as the localization problem is concerned, two different kinds of parameters can be recognized: hardware-depended and algorithm-depended. Hardware-depended parameters are mainly related to the sensors equipment. Hence, they can be properly tuned according to the particular robotic platform, e.g., σ given in eq. (4) which represents a measure of confidence related to the sensor data noise can be chosen with respect to the resolution of the laser range-finders. Algorithm-depended parameters must be tuned by simulations instead. To this end, several simulations have been carried out by ranging the parameters to find out the proper setting. The main concern was related to the robustness of these parameters with respect to environmental variations. However, according to our experience, only the dissimilarity threshold α has shown an environment dependency (experimentally validated to be set roughly to 5% of the



(a) Collaborative Localization



(b) Autonomous Localization

Fig. 9 Second scenario. Collaborative localization against autonomous localization. Plot lines' colors match robots' paths color.

environmental size), while the other parameters, e.g, crossover, mutation, tournament size and, elitism, turned out to be robust against environmental variations.

6 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 [Gasparri et al(2007)]. 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

[Cao et al(1997)] Cao YU, Fukunaga AS, Kahng AB (1997) Cooperative mobile robotics: Antecedents and directions. Autonomous Robots 4(1):7–23

[Engelson et al(1992)] Engelson S, McDermott D (1992) Error correction in mobile robot map learning. In: IEEE ICRA, Nice, France, pp 2555 – 2560

[Fox(2003)] Fox D (2003) Adapting the sample size in particle filters through kld-sampling. I J Robotic Res 22(12):985–1004

[Fox et al(2000)] Fox D, Burgard W, Kruppa H, Thrun S (2000) A probabilistic approach to collaborative multi-robot localization. Autonomous Robots 8(3):325–344

[Gao et al(2006)] Gao L, Hu Y (2006) Multi-target matching based on niching genetic algorithm. International Journal of Computer Science and Network Security 6(7):215 – 220

[Gasparri et al(2007)] Gasparri A, Panzieri S, Pascucci F, Ulivi G (2007) A spatially structured genetic algorithm over complex networks for mobile robot localization. In: IEEE ICRA, Rome, Italy

[Gasparri et al(2009)] Gasparri A, Panzieri S, Pascucci F (2009) A spatially structured genetic algorithm for multi-robot localization. Intelligent Service Robotics 2(1):31–40, DOI http://dx.doi.org/10.1007/s11370-008-0025-4

[Gerkey et al(2003)] Gerkey BP, Vaughan RT, Howard A (2003) The player/stage project: Tools for multi-robot and distributed sensor systems. In: ICAR 2003, pp 317–323

[Goldberg(1989)] Goldberg DE (1989) Genetic algorithms in search, optimization, and machine learning. Addison-Wesley

[Howard et al(2003)] Howard A, Mataric MJ, Sukhatme GS (2003) Experimental Robotics VIII, Springer Berlin / Heidelberg, chap Localization for Mobile Robot Teams: A Distributed MLE Approach, pp 146–155

[Howard et al(2004)] A. Howard, L. E. Parker, and G. S. Sukhatme, "The SDR experience: Experiments with a large-scale heterogeneous mobile robot team," in *9th International Symposium on Experimental Robotics*, Singapore, June 2004.

[Howard (2006)] A. Howard, "Multi-robot simultaneous localization and mapping using particle filters," *Int. J. Rob. Res.*, vol. 25, no. 12, pp. 1243–1256, 2006.

[Kurazume et al(1994)] Kurazume R, Nagata S, Hirose S (1994) Cooperative positioning with multiple robots. In: IEEE ICRA, vol 2, pp 1250–1257

[Mahfoud(1995)] Mahfoud SW (1995) Niching methods for genetic algorithms. PhD thesis, University of Illinois at Urbana-Champaign, Champaign, IL, USA

[Martinelli et al(2005)] Martinelli A, Siegwart R (2005) Observability analysis for mobile robot localization. In: IEEE IROS, pp 1264–1269

[Martinelli et al(2005)] Martinelli A, Pont F, Siegwart R (2005) Multi-robot localization using relative observation. In: IEEE ICRA

[Martinelli et al(2005)] A. Martinelli, F. Pont, and R. Siegwart, "Multi-robot localization using relative observations," in *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, april 2005, pp. 2797 – 2802.

[Miller et al(1995)] Miller BL, Miller BL, Goldberg DE, Goldberg DE (1995) Genetic algorithms, tournament selection, and the effects of noise. Complex Systems 9:193–212

[Mitchell(1998)] Mitchell M (1998) An Introduction to Genetic Algorithms. MIT Press, Cambridge, MA, USA

[Mourikis et al(2006)] Mourikis A, Roumeliotis S (2006) Performance analysis of multirobot cooperative localization. IEEE Trans on Robotics 22(4):666–681

[Nannen et al(2008)] V. Nannen, S.K. Smit and A. E. Eiben, "Costs and benefits of tuning parameters of evolutionary algorithms," in *Proceedings of the 10th international conference on Parallel Problem Solving from Nature.* Berlin, Heidelberg: Springer-Verlag, 2008, pp. 528–538.

[Parker(2000)] Parker LE (2000) Current state of the art in distributed robot systems. In: Proceedings of Distributed Autonomous Robotic Systems 4

[Perez et al(2008)] J. Perez, R. A. Pazos, J. Frausto, G. Rodrguez, L. Cruz, G. Mora and H. Fraire, "Self-tuning mechanism for genetic algorithms parameters, an application to data-object allocation in the web," vol. 3046, 2004, pp. 77–86.

[Rekleitis et al(1997)] Rekleitis I, Dudeck G, Milios E (1997) Multi-robot exploration of an unknown environment, efficiently reducing the odometry error. In: IJCAI 1997, pp 1340–1345

[Rekleitis et al(2002)] Rekleitis IM, Dudek G, Milios EE (2002) Multi-robot cooperative localization: a study of trade-offs between efficiency and accuracy. In: IEEE IROS, pp 2690–2696

[Rekleitis et al(2003)] Rekleitis IM, Dudek G, Milios EE (2003) Probabilistic cooperative localization and mapping in practice. In: IEEE ICRA, pp 1907–1912

[Roumeliotis et al(2002)] Roumeliotis S, Bekey G (2002) Distributed multi-robot localization. IEEE Transactions on Robotics and Autonomation 18(5):781–795

[Roumeliotis et al(2004)] Roumeliotis SI, Rekleitis IM (2004) Propagation of uncertainty in cooperative multirobot localization: Analysis and experimental results. Auton Robots 17(1):41–54

[Roumeliotis et al(2009)] E. D. Nerurkar, S. I. Roumeliotis, and A. Martinelli, "Distributed maximum a posteriori estimation for multi-robot cooperative localization," in *ICRA'09: Proceedings of the 2009 IEEE international conference on Robotics and Automation.* Piscataway, NJ, USA: IEEE Press, 2009, pp. 1375–1382.

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

[Ghidary et al(2007)] A. R. Vahdat, N. NourAshrafoddin, and S. S. Ghidary, "Mobile robot global localization using differential evolution and particle swarm optimization," in *IEEE Congress On Evolutionary Computation, CEC 2007*, 1998.

[Yuan et al(2005)] B. Yuan and M. Gallagher, "A hybrid approach to parameter tuning in genetic algorithms," vol. 2, sept. 2005, pp. 1096 – 1103 Vol. 2.