Auton Robot (2011) 31:401–417
DOI 10.1007/s10514-011-9249-9

Exploration strategies based on multi-criteria decision making for searching environments in rescue operations

Nicola Basilico · Francesco Amigoni

Received: 29 October 2010 / Accepted: 28 July 2011 / Published online: 2 September 2011

Abstract Some applications require autonomous robots to search an initially unknown environment for static targets, without any *a priori* information about environment structure and target locations. Targets can be human victims in search and rescue or materials in foraging. In these scenarios, the environment is incrementally discovered by the robots exploiting exploration strategies to move around in an autonomous and effective way. Most of the strategies proposed in literature are based on the idea of evaluating a number of candidate locations on the frontier between the known and the unknown portions of the environment according to *ad hoc* utility functions that combine different criteria. In this paper, we show some of the advantages of using a more theoretically-grounded approach, based on Multi-Criteria Decision Making (MCDM), to define exploration strategies for robots employed in search and rescue applications. We implemented some MCDM-based exploration strategies within an existing robot controller and we evaluated their performance in a simulated environment.

Keywords Exploration strategies · Frontier-based exploration · Multi-criteria decision making · Search and rescue

N. Basilico (✉) · F. Amigoni
Artificial Intelligence and Robotics Laboratory, Dipartimento di Elettronica e Informazione, Politecnico di Milano, Piazza Leonardo da Vinci 32, 20133 Milano, Italy
e-mail: basilico@elet.polimi.it

F. Amigoni
e-mail: amigoni@elet.polimi.it

1 Introduction

Situations in which mobile robots need to autonomously search for static targets in initially unknown environments are sometimes encountered in applications, like in search and rescue (Tadokoro 2010), where the targets are human victims, in some versions of map building (Thrun 2002), where the targets are landmarks, and in some versions of foraging (Scone and Phillips 2010), where the targets are materials. A particularly interesting situation is when the searching robots have no *a priori* information about the locations of the targets in the unknown environment. In this case, the problem of maximizing the number of targets found in a given time interval can be translated to the equivalent problem of maximizing the amount of area covered by robot sensors in the same time interval.

In this paper, we address this search problem in the context of search and rescue operations. *Exploration strategies* that drive the robots around a partially known environment on the basis of the available knowledge are fundamental for an effective search. The mainstream approach for developing exploration strategies is based on the idea of incrementally exploring the environment by evaluating a number of candidate observation locations according to an utility function and by selecting, at each step, the next best observation location. The candidate locations are usually on the frontier between the known and the unknown portions of the environment. Exploration strategies differ in the utility functions they use to evaluate candidate locations. These utility functions aggregate different criteria measuring different aspects of the locations. From the literature, it emerges that utility functions exploit aggregation methods that are rarely based on a theoretical ground.

In this paper, we apply a decision-theoretical tool, called *Multi-Criteria Decision Making* (*MCDM*) (Grabisch and

Labreuche 2008), to define exploration strategies for search and rescue in initially unknown environments with no information about the location of the victims. Using decision-theoretical tools, on the one hand, contributes to further strengthen the scientific foundations of robotics and, on the other hand, provides practical advantages to the definition of effective exploration strategies. Although MCDM has been preliminary applied to map building (Basilico and Amigoni 2009), we deem that its application to search and rescue represents a significant contribution since it addresses a more challenging setting for exploration strategies, where the primary objective is not to build an accurate map of the physical space but to search the environment for locating the largest number of victims in a limited amount of time. Differently from map building, search and rescue settings are strongly constrained by time and battery limitations and generally require to privilege the amount of explored area over the map quality.

Broadly speaking, we consider a multirobot system operating according to the following steps: (a) the robots perceive the surrounding environment, (b) they integrate the perceived data within a map representing the environment known so far, (c) they decide where to go next, and (d) they go there and start again from (a). We propose to use MCDM for addressing step (c), namely for defining the exploration strategy. In our experiments, we implemented the proposed approach as a modification of a publicly available controller used for the RoboCup Rescue Virtual Robots Competition (Visser and Slamet 2008). In this way, on the one hand, we can focus on the development of exploration strategies (step (c)) exploiting an already tested framework for steps (a), (b), and (d) and, on the other hand, we enable a fair comparison of our strategies with that originally used by Visser and Slamet (2008). We explicitly note that, although in multirobot exploration the evaluation of candidate observation locations is closely related to their coordinated allocation to the available robots (Gerkey and Mataric 2004), in this paper we focus only on evaluation of candidate observation locations, exploiting the task allocation method implemented in Visser and Slamet (2008).

This paper is structured as follows. The next section reviews the related works on robotic exploration. Section 3 introduces the basics of MCDM, which is applied to exploration of unknown environments for search and rescue operations in Sect. 4. Section 5 presents and discusses experimental results and Sect. 6 concludes the paper.

2 Related works

Robotic exploration can be broadly defined as a process that discovers unknown features in environments by means of mobile robots. Exploration is employed in several tasks, like map building (Thrun 2002), search and rescue (Tadokoro 2010), and coverage (Choset 2001). For example, in map building the features to be discovered can be the obstacles and the free space, while in search and rescue they can be the locations of victims or sources of danger (e.g., fires). Robots use *exploration strategies* to autonomously decide where to move in order to acquire new information and discover new features.

In this paper, we are interested in search problems in initially unknown environments with no *a priori* information about the location of the targets (e.g., victims in search and rescue). Hence, we will focus on exploration strategies employed for discovering the physical structure of environments that are initially unknown. In these scenarios, we do not know *ex ante* the complete set of the possible locations that the robots can reach. We explicitly note that, as a consequence, we cannot employ some approaches, like those proposed in Low et al. (2008) and Singh et al. (2009), which require knowledge of the set of possible observation locations. In the following, we survey a representative sample of the several exploration strategies that have been proposed in literature.

2.1 Exploration strategies for map building

Not surprisingly, most of the exploration strategies for discovering the physical structure of environments have been proposed in the context of map building. Assuming that sensors of the robots have a limited range, the mainstream approach models exploration as an incremental Next Best View (NBV) process, i.e., a repeated greedy selection of the next best observation location within the currently explored portion of the environment. Usually, at each step, an NBV system considers a number of candidate locations on the frontier between the known free space and the unexplored part of the environment (in such a way they are reachable from the current position of the robot) and selects the best one (Yamauchi 1997). The most important feature of an exploration strategy is how it evaluates candidate locations in order to select the best one.

In evaluating candidate locations, different criteria can be used. A simple one is the *distance* from the current position of the robot (Yamauchi 1997), according to which the best observation location is the nearest one. However, most works combine different criteria in more complex utility functions. For example, in Burgard et al. (2005) the cost of reaching a candidate location p is linearly combined with its expected benefit (a similar approach is used also by Stachniss and Burgard 2003). Measuring the cost as the distance $d(p)$ of p from the current location of the robot and the benefit as an estimate of the *new information* $A(p)$ acquirable from p (measured as the expected change of the entropy of the map after the measurement in p), the global utility of p is computed as:

$$u(p) = A(p) - \beta d(p), \tag{1}$$

where β balances the relative weight of benefit versus cost (authors show that choosing β within the interval [0.01, 50] does not cause significant variations in the exploration performance). Another example of combination of different criteria is reported in Gonzáles-Baños and Latombe (2002), in which distance $d(p)$ and the expected information gain $A(p)$ (measured as the amount of unexplored area potentially visible from p) of a candidate location p are combined in an exponential function

$$u(p) = A(p)e^{-\lambda d(p)} \tag{2}$$

(where λ is a parameter that weights the two criteria). In Marjovi et al. (2009), the traveling cost to reach a location is used as the main criterion for evaluating candidate locations, while the utility of the locations (calculated according to the proximity of other robots) is used only as a tie-breaker. Amigoni and Caglioti (2010) use a technique based on relative entropy to combine traveling cost and expected information gain.

In Tovar et al. (2006), several criteria are employed to evaluate a candidate location on the frontier: travelling cost, uncertainty in landmark recognition, number of visible features, length of visible free edges, rotation and number of stops needed to follow the path to the location. They are combined in a multiplicative function to obtain a global utility value. The multiplicative form guarantees that all locations with a good global utility satisfy all the criteria well.

The above strategies aggregate different criteria in utility functions that are defined *ad hoc* and that are strongly dependent on the criteria they combine. Amigoni and Gallo (2005) dealt with this problem and proposed a more theoretically-grounded approach based on multi-objective optimization, in which the best candidate location is selected on the Pareto frontier as the one closest to the ideal candidate (defined as the, usually non existing, candidate with the maximum value for all the criteria). Besides distance and expected information gain, *overlap* is also taken into account. This criterion is related to the amount of old information that will be acquired again from a candidate location. Maximizing the overlap can improve the performance of self-localization of the robot. In this work, following a similar theoretically-grounded approach, we investigate the employment of MCDM as a general method for defining exploration strategies.

2.2 Exploration strategies for search and rescue

Compared with exploration strategies for map building, less works have been devoted to the definition of exploration strategies specifically tailored for autonomous search and rescue. A work that explicitly addresses this problem by considering an utility function to evaluate candidate observation locations is Visser and Slamet (2008). Authors propose to combine the distance $d(p)$, the expected information gain $A(p)$ (see Sect. 4.1 for details about its computation), and the probability of a successful communication $P(p)$ from a candidate location p in the following utility function:

$$u(p) = \frac{A(p)P(p)}{d(p)}. \tag{3}$$

This strategy has been employed, with good results, in some RoboCup Rescue Virtual Robots Competitions.

A different approach has been investigated by Saeedi et al. (2009), who, instead of using a set of criteria to locally evaluate candidate locations, define a metric to globally evaluate the performance of an exploration strategy according to the required time (number of steps) and to the number of multiple visits for each cell of the environment (represented by a grid map). At each step of the exploration, the value for this metric is estimated and the next cell is chosen trying to maximize the expected increase of global performance.

Another approach uses artificial potential fields (Khatib 1986) built on a grid map to drive a robot towards "good" cells of the environment by following the negative gradient. In this approach, the evaluation of candidate locations is implicitly encoded in the potential field generation technique. For instance, in Wirth and Pellenz (2007) each cell c is evaluated with respect to every frontier cell f. This is done by computing a cost value depending both on the length and on the safety (vicinity to obstacles) of the path from c to f (the two criteria are combined with a weighted sum). Cell c is then assigned a value representing its lowest cost over all the frontiers f. At each step, the robot travels to the cell with the lowest cost among those adjacent to its current cell. In order to maintain visibility over landmarks, authors show how the safety evaluation procedure for a path can be adapted to prevent the robot from moving too far from obstacles (and consequently incurring in a difficult localization). The principle behind this step (select a frontier that ensures a good localization) is similar to that discussed for the overlap criterion employed by Amigoni and Gallo (2005). A work following a similar approach is Rasche et al. (2010), where the exploratory behavior of a team of UAVs (Unmanned Aerial Vehicles) is combined with the pursuit of goal locations. The robots follow the negative gradient of a potential field computed for each cell according to the traveling cost discounted by a factor that measures the attractiveness of unexplored zones of the environment.

Several works adopt bio-inspired methods to define exploration strategies for search. For example, in Scone and

Phillips (2010) a strategy based on foraging is adopted. Robots (animals) search for victims (food) within a number of rooms (patches) composing the environment. Applying results from biological foraging literature to a dataset of exploration runs, an optimal residence time for finding victims is calculated for each room. This time is then exploited to drive exploration.

Communication between robots is an important aspect in search and rescue operations conducted with multiple robots. If collecting perceived data at a base station is fundamental to provide first responders with the current state of the search, it can be difficult on disaster sites. Some works explicitly considered this aspect in defining exploration strategies for search and rescue. An already discussed example is Visser and Slamet (2008), where the probability of a successful communication is introduced as a criterion in an utility function (the approach has been extended by de Hoog et al. 2009 to define a role-based exploration where robots can act either as explorers or as relayers of data). Scone and Phillips (2010) deal with the problem of limited communication, allowing robots to exchange information exploiting opportunistic links. In Kleiner et al. (2006) robots evaluate frontiers according to the utility function (1), while data exchange and coordination are achieved through RFID tags autonomously distributed in the environment. An approach based on periodic communication to coordinate robots has been proposed by Hollinger and Singh (2010).

Sometimes *a priori* information about victim location is available to first responders. In these situations, exploiting such knowledge in the definition of the exploration strategy could be desirable. In Calisi et al. (2007), a formalism based on Petri nets is used to consider information about the victim distribution (e.g., if they are uniformly spread or concentrated in clusters) to improve the search. The work is extended in Calisi et al. (2008) where high level contextual knowledge is integrated in the exploration strategy. Lin and Goodrich (2009) propose a technique to plan a search path that accounts for a given probability distribution of victims' locations.

Finally, some interest has been devoted to semi-autonomous exploration strategies, where the human operator plays a supervision role. An example is reported in Nevatia et al. (2008), where an exploration strategy based on (1) permits the interaction with a human operator via a graphic user interface.

Since in this paper we are assuming fully autonomous operations and no *a priori* information about victim locations, these last works are not directly comparable with our approach. However, in our experimental activity we will compare exploration strategies developed with our approach with that proposed by Visser and Slamet (2008), which is explicitly devoted to search and rescue operations.

3 Multi-criteria decision making

When designing an effective exploration strategy for exploring initially unknown environments, the main challenge is to achieve a good global (long-term) performance by means of local (short-term) decisions that are made by the robots on the basis of partial available knowledge. In our scenario, the partial knowledge is given by the current map built by the robots and short-term decisions are made by evaluating a number of alternatives, i.e., candidate observation locations on the frontiers between the explored and unexplored space, and by selecting the best one. The "goodness" of an observation location can be measured with respect to multiple criteria that depend on the goal the system has to achieve, as we have seen in the previous section. In principle, the number of criteria that can be considered is unlimited and can change from rescue operation to rescue operation and within the same operation. As the tasks the robots perform become more complex (think, for example, of an exploring robot that has to find victims, localize fire sources, communicate with a base station, and so on), this number is likely to increase.

In this work, we explicitly consider the evaluation of candidate locations as a multi-objective (or multi-criteria) optimization problem. Let us formalize the problem. We have a set C of candidate locations among which we want to choose the "best" one. We denote the set of n criteria considered in the evaluation process as $N = \{1, 2, \dots, n\}$. Given a candidate $p \in C$ we denote with $u_i(p) \in I$ its *utility* with respect to criterion $i \in N$, where $I \subseteq \mathbb{R}$ represents the set of possible utility values. Note that we assume that all utilities have values over the same set I. The larger the utility $u_i(p)$, the better candidate location p satisfies criterion i. Each candidate p can be associated to a vector of n elements, namely its utilities, $u_p = (u_1(p), u_2(p), \dots, u_n(p))$. The problem of selecting the "best" candidate observation location comes down to the problem of selecting the optimal candidate location p^* from C.

Dealing with this multi-criteria scenario, the optimality of candidates involves the concept of *Pareto frontier*. Formally, the Pareto frontier of C can be defined as the largest subset $P \subseteq C$ such that, for every $p \in P$, there is not any candidate $q \in C$ with $u_i(q) > u_i(p)$ for all $i \in N$. A candidate $q \in C \setminus P$ is said to be *Pareto-dominated* and can be safely discarded, since at least a preferable candidate is guaranteed to exist in P. Therefore, choosing a candidate on the Pareto frontier P is a fundamental requirement to select a "good" candidate. The actual selection is performed via a *global utility function* $u(p) = f(u_p) = f(u_1(p), u_2(p), \dots, u_n(p))$ that combines together utilities in an aggregate value (well-known examples are the arithmetic and weighted mean). Since computing the Pareto frontier P can be computationally expensive (especially when the number of candidates grows), the selection is usually done by looking directly at the initial set C, namely $p^* =$

$\arg\max_{p \in C} f(u_p)$. It can be easily shown that if $f()$ is a non-decreasing function in every one of its n arguments, then p^* is guaranteed to be on the Pareto frontier. As the previous section shows, the mainstream approach followed in literature to define global utility functions is to combine a pre-determined number of criteria in an *ad hoc* form. Despite it is not explicitly mentioned, almost all these methods are Pareto optimal, since a non-decreasing global utility function is a "natural" choice.

In the following sections we describe Multi-Criteria Decision Making (MCDM) as a general method for defining global utility functions and we discuss some of its advantages and properties that make it a valid tool for exploration strategies.

3.1 Combining criteria with the Choquet integral

We introduce and motivate the proposal of MCDM by considering the important aspect of the dependency between criteria, that is often neglected by simple global utility functions. Criteria that are used to evaluate candidate locations are not always independent. For example, think of criteria that estimate the same feature using different methods, like two criteria that estimate the distance of a candidate location from the current position of the robot according to the Euclidean and to the Manhattan distance. Intuitively, when combining them into a global utility function, their overall contribution to the global utility of a candidate location should be less than the sum of their individual ones. In this case, a *redundancy* relation holds between criteria. A dual situation occurs when two or more criteria are very different and, in general, can be hardly optimized together. In this case, a *synergy* relation holds between criteria, and their overall contribution should be considered larger than the sum of the individual ones. An example involves the estimated information gain and the overlap. These criteria can be considered synergic, since large utilities for both are very difficult to achieve by a single candidate, and candidates that satisfy both criteria reasonably well should be preferred to candidates that satisfy them in an unbalanced way. In order to consider these issues, we need a way to define a global utility function that accounts for redundancy and synergy between criteria when combining them. MCDM provides a general aggregation method which can deal with this and with other aspects and that exploits the *Choquet integral* to compute global utilities (Grabisch and Labreuche 2008). Let us introduce it.

We first introduce a (total) function $\mu : \mathcal{P}(N) \to [0, 1]$ ($\mathcal{P}(N)$ is the power set of set N) with the following properties:

- $\mu(\{\emptyset\}) = 0$,
- $\mu(N) = 1$,
- if $A \subseteq B \subseteq N$, then $\mu(A) \leq \mu(B)$.

That is, μ is a normalized *fuzzy measure* on the set of criteria N that will be used to associate a weight (or an importance) to each group of criteria. The weights specified by the definition of μ describe the dependency relations that hold for each group of criteria. Criteria belonging to a group $G \subseteq N$ are said to be redundant if $\mu(G) < \sum_{i \in G} \mu(i)$, synergic if $\mu(G) > \sum_{i \in G} \mu(i)$, and independent otherwise.

The global utility $f(u_p)$ for a candidate p is computed as the discrete Choquet integral with respect to the fuzzy measure μ using p's utilities:

$$f(u_p) = \mathcal{C}(u_p) = \sum_{j=1}^{n} (u_{(j)}(p) - u_{(j-1)}(p))\mu(A_{(j)}), \quad (4)$$

where $(j) \in N$ indicates the j-th criterion according to an increasing ordering with respect to utilities, i.e., after criteria have been permutated to have, for candidate p,

$$u_{(1)}(p) \leq \cdots \leq u_{(n)}(p) \leq 1.$$

It is assumed that $u_{(0)}(p) = 0$. Finally, the set $A_{(j)}$ is defined as

$$A_{(j)} = \{i \in N | u_{(j)}(p) \leq u_i(p) \leq u_{(n)}(p)\}.$$

Using $\mathcal{C}(u_p)$ to compute global utilities allows to consider importance of criteria and their mutual dependency relations and brings some interesting properties on the stage.

3.2 Some properties of MCDM

In this section, we discuss a number of properties of the proposed MCDM approach. A first general feature of the Choquet integral is that, differently from *ad hoc* global utility functions, it can be applied to any number of criteria. Indeed, rigorously speaking, $\mathcal{C}()$ as defined in (4) is not an aggregation function, for which the number of arguments should have been fixed *a priori*, but an *aggregation operator*. An aggregation operator is a collection of aggregation functions, one for each number n of criteria to be combined. For example, the arithmetic and weighted means are aggregation operators since they basically specify an aggregation technique for every possible number of criteria, while global utility functions like (2) and (3) are aggregation functions suitable only for the set of criteria they have been tailored for. In this sense, we can say that an aggregation operator is more general than an aggregation function. An obvious advantage of using an aggregation operator instead of an aggregation function is the increased flexibility, namely adding and removing criteria can be accomplished preserving the way in which they are combined. As we will discuss in next sections, this feature enables easy refinements of the exploration strategies and facilitates some experimental activities such as assessing the impact of removing or including a criterion.

$\mathcal{C}()$ enjoys several other properties (Grabisch and Labreuche 2008). Here, we briefly discuss some properties that are significant in connection with the definition of exploration strategies and that characterize MCDM as a suitable approach to define global utility functions.

Increasing monotonicity in each argument

For all $u_p, u'_p \in I^n$,

- if $\forall i \in N,\ u_i(p) \le u'_i(p)$, then $\mathcal{C}(u_p) \le \mathcal{C}(u'_p)$,
- if $\forall i \in N,\ u_i(p) < u'_i(p)$, then $\mathcal{C}(u_p) < \mathcal{C}(u'_p)$.

This property can be exploited to guarantee that the maximization of $\mathcal{C}()$ over the set of candidate locations C will select a Pareto optimal candidate. As we discussed before, almost all aggregation functions proposed in literature for exploration strategies satisfy this property.

Stability for linear transformations

For all $u_p \in I^n$ and $r, s \in \mathbb{R}$ with $r > 0$ such that, for all $i \in N$, $ru_i(p) + s \in I$, it holds that

$$\mathcal{C}(ru_1(p)+s, ru_2(p)+s, \dots, ru_n(p)+s) = r\mathcal{C}(u_1(p), u_2(p), \dots, u_n(p)) + s.$$

This property states the independence of the particular scale in which utilities are measured (up to a linear transformation). If this property holds, the assessment of how better a candidate is with respect to another candidate is independent of the scale used to measure utilities. In this paper we assume, without any loss of generality, that utilities have values in $I = [0, 1]$ (any other common scale would have been equivalent). In general, this property is rarely satisfied by aggregation functions proposed in literature, where often criteria are measured with respect to different scales and combined without any normalization (see, for example, Gonzáles-Baños and Latombe 2002 and Visser and Slamet 2008).

Continuity

Given n, the corresponding aggregation function $\mathcal{C}()$ is continuous on I^n. This property prevents the global utility to exhibit irregular variations with respect to small changes of the utility values that are aggregated. This property is satisfied also when the global utility is computed by adopting exponential or fractional functions (see (2) and (3)).

Idempotence

If, for a given p, all $u_i(p) = u \in I$, then

$$\mathcal{C}(u_1(p), u_2(p), \dots, u_n(p)) = \mathcal{C}(u, u, \dots, u) = u.$$

This property assures a sort of consistency, namely, if all the criteria are satisfied with the same degree u, then the global utility is u. This property is rarely exhibited by the aggregation functions used in literature, with the drawback that the particular form in which criteria are combined can introduce a bias in the evaluation, for example by implicitly giving more importance to some criteria to the detriment of others.

3.3 Generality of MCDM

Another important advantage of MCDM is its generality. Indeed, different aggregation operators turn out to be particular cases of the Choquet integral, up to a proper choice of weights for the fuzzy measure μ. For instance, a class of aggregation operators that can be expressed with the Choquet integral are *weighted means*. A weighted mean is defined as $\sum_{i=1}^n w_i u_i(p)$ where w_i is the weight of criterion i and $\sum_{i=1}^n w_i = 1$. This aggregation operator can be obtained from Choquet integral by setting $\mu(\{i\}) = w_i$ for all $i \in N$ and by constraining μ to be additive:

$$\mu(S) = \sum_{i \in S} w_i \quad \forall S \in \mathcal{P}(N).$$

Note that additivity of μ reflects independence between criteria, namely joint contributions are exactly the sum of marginal ones. Therefore, weighted means should be considered suitable when such independence between criteria actually holds. Moreover, the arithmetic mean and the k-th criterion projection (namely, considering only a criterion) can be obtained as further particular cases of weighted means by imposing $w_i = 1/n\ \forall i \in N$ and $w_k = 1$, $w_i = 0\ \forall i \in N \setminus \{k\}$, respectively. In the context of exploration, this means that the strategy proposed in Burgard et al. (2005) and based on (1) can be viewed as a special case of MCDM-based exploration strategies. Moreover, also the global utility function proposed in Marjovi et al. (2009) can be viewed as a special case of MCDM, basically being a k-th criterion projection.

A second class of aggregation operators that are special cases of the Choquet integral is composed of *ordered weighted means*. An ordered weighted mean is defined as $\sum_{j=1}^n w_j u_{(j)}(p)$ (i.e., a weighted mean in which w_j is the weight of the j-th criterion according to an increasing ordering of utilities). An ordered weighted mean aggregation operator can be obtained from the Choquet integral by setting $\mu(\{j\}) = w_j$ for all $j \in N$ and by defining $\mu(S)$ according to:

$$\mu(S) = \sum_{j=n-|S|+1}^{n} w_j \quad \forall S \in \mathcal{P}(N).$$

Some further particular cases of ordered weighted means that can be modeled with a proper choice of weights w_j are the minimum and maximum (when $w_1 = 1$ and $w_n = 1$, respectively), the median (when $w_{\frac{n}{2}} = w_{\frac{n}{2}+1} = 0.5$ and n is even or when $w_{\frac{n+1}{2}} = 1$ and n is odd), and the arithmetic mean excluding the two extremes (when $w_1 = w_n = 0$ and $w_j = \frac{1}{n-2}\ \forall j \in N \setminus \{1, n\}$). This shows the possibility offered by MCDM of obtaining completely different global utility functions (and, as a consequence, different behaviors of the robots) by simply setting weights μ. In this sense, we say that MCDM constitutes a general approach for defining exploration strategies.

4 MCDM-based exploration strategies for search and rescue

We apply the proposed MCDM approach to define exploration strategies for search and rescue, where mobile robots are deployed in an initially unknown environment with the goal to explore it and search for human victims within a limited amount of time. As discussed in Sect. 1, this applicative domain offers a challenging scenario to test exploration strategies in searching fixed targets with no information about their locations. Using MCDM does not guarantee *per se* to get a good exploration strategy. MCDM provides a good formulation for defining exploration strategies composing criteria measured by utilities defined by the designer.

We implemented MCDM-based exploration strategies in an existing robot controller for search and rescue applications. We looked at the participants to the RoboCup Rescue Virtual Robots Competition where different teams compete in developing simulated robotic platforms operating in Urban Search and Rescue scenarios simulated in USARSim (Carpin et al. 2007). From an analysis based on availability of code and performance obtained in the competition, we selected the controller developed by the Amsterdam and Oxford Universities (Amsterdam Oxford Joint Rescue Forces, AOJRF)[1] for the 2009 competition (Visser et al. 2009). The reasons for implementing MCDM-based exploration strategies in an existing controller are that we can focus only on the exploration strategies, exploiting existing and tested methods for navigation, localization, and mapping and that we have a fair way to compare our exploration strategies with that originally used in the controller. In the following we describe the original controller and how we modified it to implement MCDM-based strategies.

4.1 The AOJRF controller

In this section, we describe some of the controller's features that are relevant to the scope of this paper (please refer to Visser and Slamet 2008 for a complete description).

[1] http://www.jointrescueforces.eu/.

The controller manages a team of robots. The robotic platform used is a Pioneer P3AT. The map of the environment is maintained by a base station, whose position is fixed in the environment, and to which robots periodically send data. The map is two-dimensional and represented by three occupancy grids. The first one is obtained from maximum-range scans (typically 20 meters) and constitutes the *free area*, i.e., the area which is believed to be free but not yet safe. The second one is obtained with a small-range (typically 3 meters) scanner and constitutes the *safe area*, i.e., the area where the robots can safely move. Moreover, a representation of the *clear area* is also maintained as a subset of the safe area that has been checked for the presence of victims (this task is accomplished with simulated sensors for victim detection). Given a map represented as above, a set of boundaries between safe and free regions are extracted and considered as frontiers. For each frontier, the middle point is considered as a candidate location to reach. The utility of a candidate location p is evaluated by combining the following criteria:

- $A(p)$ is the amount of the free area beyond the frontier of p computed according to the free area occupancy grid;
- $P(p)$ is the probability that the robot, once reached p, will be able to transmit information (such as the perceived data or the locations of victims) to the base station (whose position is *a priori* fixed within the environment), this criterion depends on the distance between p and the base station;
- $d(p, r)$ is the distance between p and current position of robot r, this criterion can be calculated with two different methods: $d_{EU}()$, using the Euclidean distance, and $d_{PP}()$, using the exact value of the distance returned by a path planner.

Given these criteria, the global utility for a candidate p is calculated using function (3). We will refer to the exploration strategy using this global utility function as the "AOJRF strategy".

The allocation of candidate locations to robots is performed with the following algorithm, which is executed by each robot independently, knowing (from the base station) the current map and the positions of other robots (Visser and Slamet 2008):

1. compute the global utility $u(p, r)$ of allocating each candidate p to each robot r using (3) where $d(p, r)$ is calculated using the Euclidean distance $d_{EU}()$ (namely using an underestimate of the real distance):

$$u(p, r) = \frac{A(p)P(p)}{d_{EU}(p, r)},$$

2. find the pair (p^*, r^*) such that the previously computed utility is maximum, $(p^*, r^*) = \arg\max_{p,r} u(p, r)$,

3. re-compute the distance between p^* and r^* using $d_{PP}()$ with the path planner (namely considering the real distance) and update the utility of (p^*, r^*) using such exact value instead of the Euclidean distance:

$$u(p^*, r^*) = \frac{A(p^*)P(p^*)}{d_{PP}(p^*, r^*)},$$

4. if (p^*, r^*) is still the best allocation, then allocate robot r^* to location p^*, otherwise go to step 2,
5. eliminate robot r^* and candidate p^* and go to step 2.

The reason behind the utility update of step 3 is that computing $d_{PP}()$ requires a considerable amount of time. Doing this for all the candidate locations and all robots would be not affordable in the rescue competition, since a maximum exploration time of 20 minutes is enforced.

4.2 Developing MCDM-based strategies

We now describe the definition of MCDM-based strategies and their inclusion in the original controller.

To define a MCDM-based exploration strategy, a set of weights for μ should be determined. This can be a particularly tricky problem. The designer should consider the applicative domain and, by assigning weights to groups of criteria, encode a particular decision policy with which the robot will evaluate and select candidate locations. We explicitly remark that setting parameters is a trait common to every approach for exploration strategies. For example, in the strategies defined by (1) and (2), this problem arises when choosing values for β and λ, respectively. Similarly, a more theoretically grounded approach like MCDM does not come with "out-of-the-box" weights to use. However, differently from *ad hoc* approaches, MCDM provides a better framework for the problem of choosing weights, giving the designer a more clear idea of the meaning of parameters. We now illustrate the exploration strategies we defined by manually setting the weights and later we come back to the definition of weights.

The first MCDM-based strategy we propose adopts the same criteria of the AOJRF strategy (i.e., A, P, and d, as described above), but combines them with the MCDM approach. Basically, we replace function (3) with function (4), with the weights reported in Table 1 (top). We call this the "MCDM strategy". The MCDM strategy assigns more importance to A than to P and d (see Table 1 (top)), pushing the robot to discover new areas, even covering long distances or risking a loss of communication. The joint contribution of d and P is inhibited by establishing redundancy between them. On the other side, a synergy holds between d and A, privileging locations satisfying these criteria in a balanced way.

Table 1 Weights used for the MCDM-based strategies

criteria	$\mu()$	criteria	$\mu()$
MCDM			
A	0.5	A, d	0.95
d	0.3	A, P	0.7
P	0.2	d, P	0.4

criteria	$\mu()$	criteria	$\mu()$
MCDMb			
A	0.4	d, P	0.25
d	0.25	d, b	0.35
P	0.1	P, b	0.25
b	0.25	A, d, P	0.75
A, d	0.75	A, d, b	0.9
A, P	0.5	A, P, b	0.75
A, b	0.65	d, P, b	0.45

criteria	$\mu_1()$	$\mu_2()$
MCDMw		
A	0.6	0.4
d	0.1	0.5
P	0.3	0.1
A, d	0.8	0.95
A, P	0.9	0.5
d, P	0.3	0.5

Table 2 Interaction indexes and Shapley's values for μ_1

	Interaction			Shapley
	A	d	P	
A	–	0.1	0	0.65
d	0.1	–	−0.1	0.1
P	0	−0.1	–	0.25

To apply MCDM, utilities have to be normalized to the chosen common scale $I = [0, 1]$. The robot decision at any step depends only on C and not on previous decisions and previous sets of candidate locations. Hence, we use a normalization that linearly scales the values of utility with respect to a criterion among the current set of candidate locations C. For example, given a robot r, the utility of a candidate p related to the distance $d()$ is normalized using $u_d(p, r) = 1 - (d(p, r) - \min_{q \in C} d(q, r))/(\max_{q \in C} d(q, r) - \min_{q \in C} d(q, r))$. This poses a problem for normalizing the updated utility in step 3, since it would require to determine the path for every candidate location, making the 20 minutes limit too strict to achieve an acceptable performance (recall that $d_{PP}()$ is computationally expensive). To deal with this problem we use the following procedure in step 3: once computed $d_{PP}(p^*, r^*)$, we normalize it by using the previously calculated values $d_{EU}(p, r^*)$ for other candidates $p \in C$.

The second MCDM-based strategy we propose is intended to show the flexibility of MCDM in adding a new criterion, i.e., the robot's battery remaining charge b. Explicitly considering the battery can improve exploration by preventing the robot from making decisions it cannot complete (e.g., selecting a location not reachable with the residual energy). To compute $b()$ we need an estimate of the energy spent for reaching p. We consider a very simple model in which the power consumption is related to the time required for reaching p. In order to estimate the time needed to reach a location p we consider the path the robot should follow in terms of linear segments and rotations. By approximating the linear and angular velocities of the robot as constants (0.22 m/s and 0.2 rad/s, respectively), we can derive estimates of the time $b(p, r)$ needed by robot r to reach p. Obviously, the smaller $b(p, r)$ the better (and the larger) $u_b(p, r)$. Notice that $u_b(p, r)$ (related to the time needed by the robot r to reach p) and $u_d(p, r)$ (related to the distance between the robot r and p) show an evident dependency relation given by the fact that long traveling distances often correspond to long times. However, despite this similarity, including b in the set of criteria can, to some extent, provide more informed decisions since it captures also the difficulty for covering a path, which generally is not captured by d (consider, for example, short but winding paths that could require lot of time and battery). Modeling a redundancy relation between these two criteria is the proper way to include both of them in the decision-making process without unbalancing decisions toward the common principle embedded in b and d. We denote the strategy including b as "MCDMb strategy", whose weights are reported in Table 1 (middle). As it can be seen, the weight assigned to the set $\{d, b\}$ is lower than the sum of weights of b and d.

We also show how MCDM can be used for defining different *behaviors* in exploration. Broadly speaking, a behavior defines the preferences according to which a robot selects observation locations. Given a set of criteria, a behavior is associated to the particular set of weights of those criteria. By changing the weights during exploration, we can switch between different behaviors, varying the criteria's importance that drive robot's decisions. This technique allows us to adapt the exploration strategy to different situations. Hence, we define a third MCDM-based strategy, called the "MCDMw strategy", whose weights are reported in Table 1 (bottom). This strategy encloses two different behaviors, given by the sets of weights μ_1 and μ_2, defined over the original set of criteria of the MCDM strategy (i.e., A, P, and d, as described above). In addition, we define the following policy for switching through behaviors. The weights defined by μ_1 are used during the first 10 minutes of search while those defined by μ_2 are used during the last 10 minutes. The first set of weights encodes an aggressive behavior oriented towards the maximization of the new area. This behavior is reasonable during the first part of the search when a long remaining time is left and the robot can privilege the amount of new area even if long paths have to be followed. Differently, the second set of weights induces a more conservative behavior. This behavior accounts for the fact that remaining time is short and gives more importance to distance, preferring closer locations (note, for instance, that $\mu_1(d) = 0.1$ while $\mu_2(d) = 0.5$).

For defining the above exploration strategies, weights have been chosen empirically. Although it worked in our case, given the limited number of criteria in our setting, manual setting of weights is not an affordable option when the number of criteria is larger. Different methods have been proposed to set weights for MCDM in a semi-automated way, without having to manually specify all the $2^n - 2$ needed values. We now discuss one of these methods; for a thorough survey, please refer to Grabisch et al. (2008).

A more principled selection of weights starts from two features that characterize a set of weights: the *overall importance* of each criterion i and the *interaction* between any pair of criteria i and j. The overall importance of a criterion i is quantified by its *Shapley's value* $\phi_\mu(i) \in [0, 1]$, defined as the average marginal contribution that criterion i brings to each subset of criteria not including it:

$$\phi_\mu(i) = \sum_{T \subseteq N \setminus \{i\}} \frac{(|N| - |T| - 1)!|T|!}{|N|!} (\mu(T \cup \{i\}) - \mu(T)).$$

The higher $\phi_\mu(i)$, the more important criterion i in the decision process. The interaction between two criteria i and j is similarly defined by the *Shapley's interaction index* $I_\mu(i, j) \in [-1, 1]$:

$$I_\mu(i, j) = \sum_{T \subseteq N \setminus \{i,j\}} \frac{(|N| - |T| - 2)!|T|!}{(|N| - 1)!} M(T, i, j),$$

where

$$M(T, i, j) = [\mu(T \cup \{i, j\}) - \mu(T \cup \{i\})] - [\mu(T \cup \{j\}) - \mu(T)]$$

is the marginal interaction between i and j in the set of criteria T. If $I_\mu(i, j) < 0$, then i and j are redundant and $I_\mu(i, j)$ measures the "intensity" of this relationship with the minimum value in -1 representing full redundancy (namely, the two criteria i and j are identical). Symmetric considerations hold for $I_\mu(i, j) > 0$ and synergy relationship. $I_\mu(i, j) = 0$ reflects mutual independence between i and j.

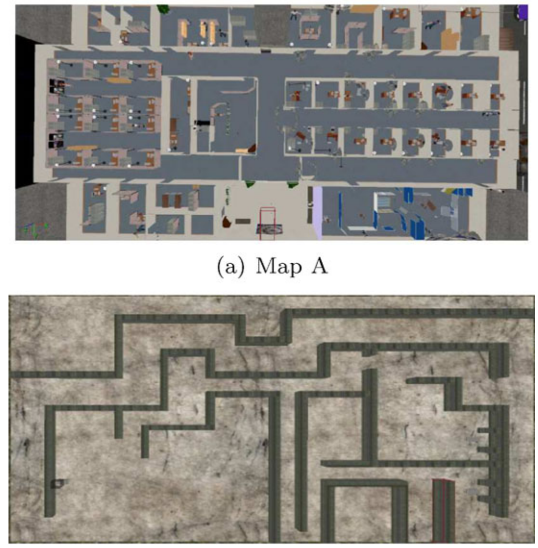
A simple method to get a set of weights for μ is to define constraints over some of these indexes, by specifying bounds or exact values, and to solve a corresponding linear program to find a feasible set of weights (see Grabisch et al. 2008 for details). In this way, the designer is required to specify a limited number of values for indexes, quantifying the overall importances and interactions of the most important criteria. (By using Shapley's interaction indexes, we can model only dependencies between pairs of criteria. However, as discussed by Grabisch et al. (2008), this is a common assumption for many applications.)

As an example of application of the above semi-automated method for setting weights, we consider the case of μ_1 in MCDMw strategy (Table 1 (bottom)). The weights have been computed by imposing Shapley's values and interaction indexes reported in Table 2. Since μ_1 is intended to show an aggressive behavior, an high overall importance (0.65) is chosen for the criterion A while criteria d and P have relatively small Shapley values (0.1 and 0.25, respectively). Interaction indexes reflect a slight synergy between A and d (0.1) and a slight redundancy between d and P (-0.1). Mutual independence between A and P was imposed by setting to 0 the corresponding interaction indexes. The modeling of all these constraints and the resolution of the corresponding linear program to find feasible weights have been performed exploiting the Kappalab R package.[2] The weights returned by the software are those reported in Table 1 (bottom) for μ_1.

5 Experimental evaluation

We conducted our experimental activities in the same setting of the RoboCup Rescue Virtual Robots Competition, in which robots and environments are simulated in USARSim (Carpin et al. 2007) and a time limit of 20 minutes is imposed to exploration.

We first present experiments that evaluate the performance of the MCDM strategy when compared with other strategies. We consider the AOJRF strategy (corresponding to (3)), the WS strategy (corresponding to (1) with $\beta = 1$), and the DIST strategy, by which locations are selected simply by minimizing d (i.e., always choosing the nearest location). AOJRF and WS strategies are based on continuous and increasingly monotonic aggregation functions. These two strategies guarantee a Pareto optimal selection, however AOJRF strategy lacks in flexibility since including further criteria would require to re-define the aggregation technique, while WS can be considered as a special case of MCDM-based strategies (see Sect. 3.3). Nevertheless, AOJRF and WS strategies have been proved to achieve good results in


(a) Map A

(b) Map B

Fig. 1 The maps used for tests

practice (Visser et al. 2009), therefore, by comparing the MCDM strategy with them, we aim at deriving insights on how performance changes when using a more theoretically-grounded way to define utility functions. DIST is a very simple strategy that can be viewed as a particular case of MCDM-based strategies. Indeed, it can be obtained by restricting the set of criteria to the singleton d (see Sect. 3.3). By comparing MCDM and DIST we aim at confirming that making more informed local decisions actually results in a better global performance.

We considered teams of one or two robots deployed in the two indoor environments of Fig. 1 that show different characteristics. Map A is cluttered and composed of corridors and many rooms, while Map B is characterized by the presence of open spaces. A configuration is defined by an environment, a team of robots deployed in it, and the exploration strategy adopted. For each configuration, we executed 10 runs (with randomly selected starting locations for the robots) of 20 minutes each. We assess performance by measuring the amount of free, safe, and clear area at each minute of the exploration (this is one of the most important metrics used in the RoboCup Rescue Virtual Robot Competition (Balaguer et al. 2009)). In Fig. 2 we report the results obtained for free, safe, and clear area with one robot in Map B. Each graph shows how the average mapped area varies with time using different strategies (each point is the average over 10 runs). Free area turned out to not be notably significant in showing differences between the compared strategies. As it can be seen (Fig. 2(a)), all the strategies (with a slight exception of DIST), due to the use of maximum-range scan (typically 20 m), show the same fast grow with respect to time. (Recall that this area is considered to define boundary regions where frontiers lie and to provide an estimate of

[2] http://ikojadin.perso.univ-pau.fr/kappalab/.

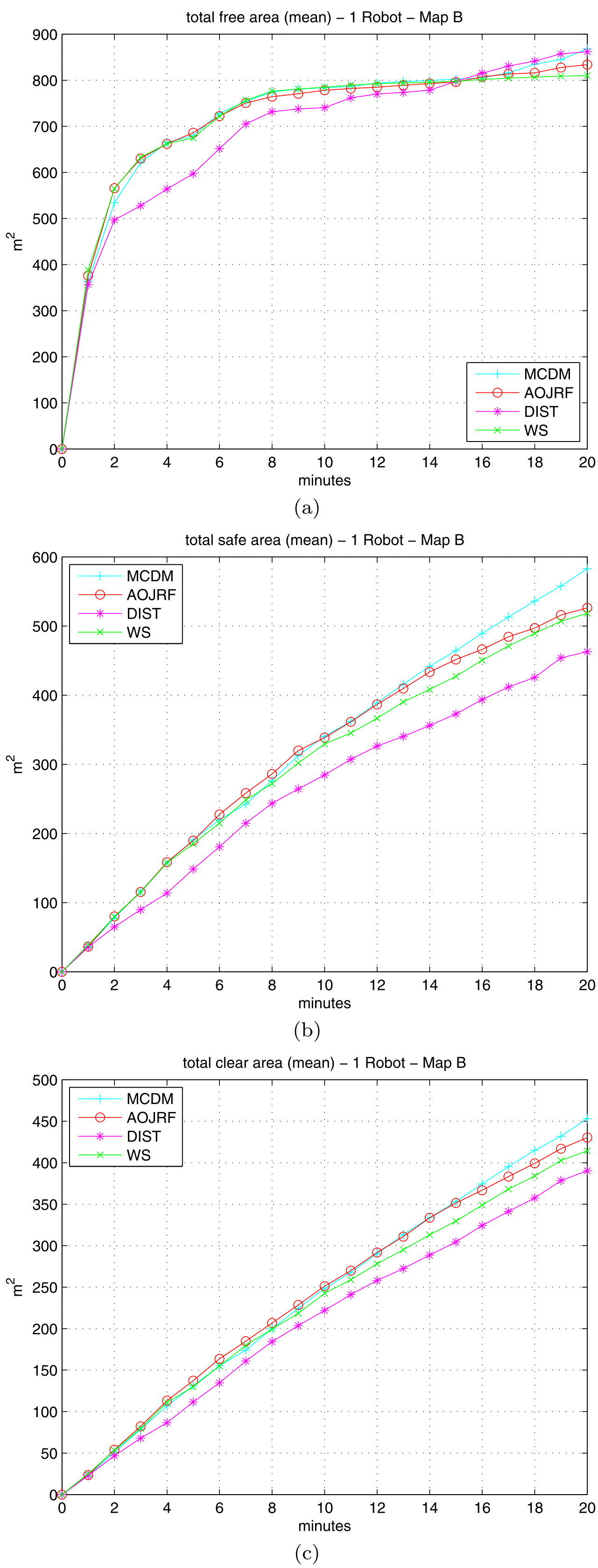


Fig. 2 Results for free (**a**), safe (**b**), and clear (**c**) area

new area beyond a frontier (see Sect. 4.1).) Hence, we do not consider it as a significant measure of the mapped area and to assess performance of exploration. Moreover, in all experiments clear area showed a trend very similar to that of safe area (compare (b) and (c) of Fig. 2). For this reason, in the following we report only data on safe area, which we consider the most representative metric for performance assessment (most of the following considerations hold also for free and clear area).

Figures 3 and 4 show the results of the experiments with a team of one and two robots in the two environments. Histograms compare the number of runs in which a strategy obtained the largest amount of safe area at the end of the 20 minutes exploration. Graphs show the trend of average safe area with respect to time. Histograms and the graphs convey two different and complementary kinds of information about experimental data. The former ones show cumulative data, while the latter ones show average data. For example, the top histogram of Fig. 3 (right) shows that if runs were competitions, MCDM would win 8 times out of 10, while the bottom graph of Fig. 3 (right) shows that the average safe area mapped by MCDM over the 10 runs is only slightly larger than that mapped using other exploration strategies. Although less significant than graphs showing average data, histograms are closer to the way exploration strategies would be evaluated in the RoboCup Rescue Virtual Robots Competition.

The MCDM strategy discovered the largest area in the majority of runs, outperforming (on average) other strategies. According to an ANOVA test (Pestman 1998) and considering a threshold for significance p-value < 0.01, the differences between the average total safe area mapped in 20 minutes (in Map A) are statistically significant between DIST and each one of the other three strategies (the p-values of the comparisons between DIST and MCDM, AOJRF, and WS, are 1.1×10^{-7}, 6.16×10^{-6}, and 2.17×10^{-5}, respectively). Differences between MCDM and AOJRF (p-value $= 0.159$), MCDM and WS (p-value $= 0.236$), and AOJRF and WS (p-value $= 0.734$) are not statistically significant in Map A. In Map B, the MCDM strategy shows a statistically significant difference when compared to DIST (p-value $= 6 \times 10^{-4}$) and AOJRF (p-value $= 9.4 \times 10^{-3}$), while the difference between MCDM and WS is not statistically significant (p-value $= 0.082$). Figure 5 illustrates these data for a team of two robots. These results reflect an interesting insight associated to the different characteristics of the two environments. Map A is cluttered and, exploring it, the robots deal with a relatively large number of frontiers among which to choose (30 candidate locations on average at each step with one robot and 40 with two robots). Map B is characterized by open spaces, resulting in a smaller number of candidate frontiers (5 candidate locations on average at each step with one robot and 8 with two robots). However, despite their large number, frontiers in Map A are very

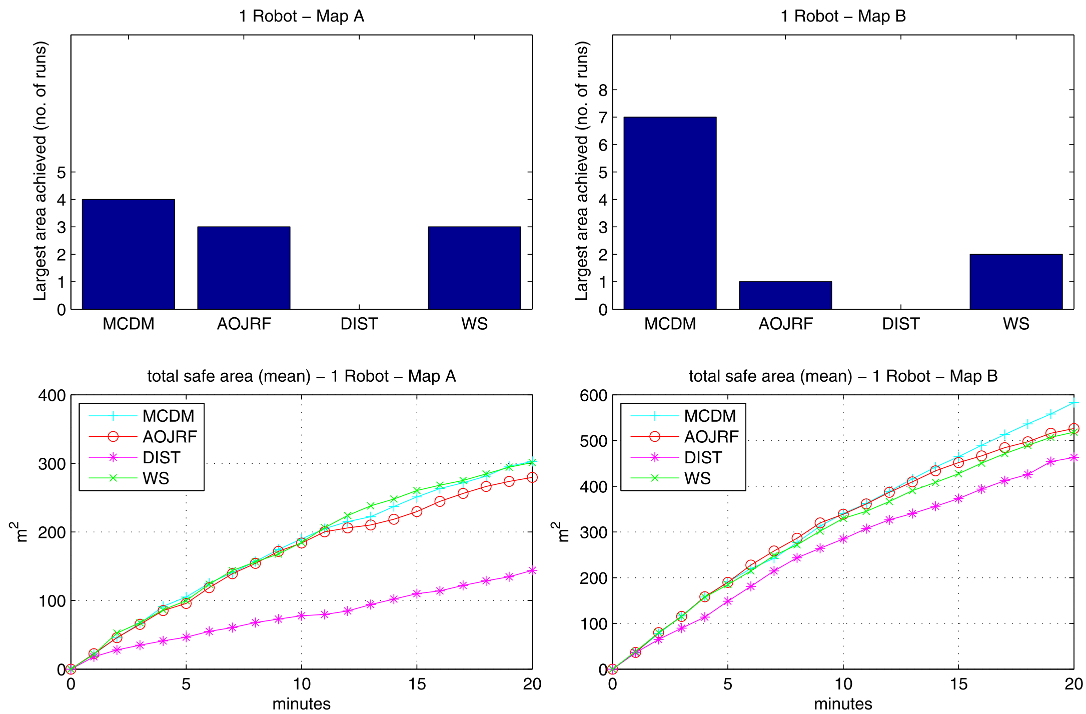


Fig. 3 Comparison between MCDM and other exploration strategies with one robot

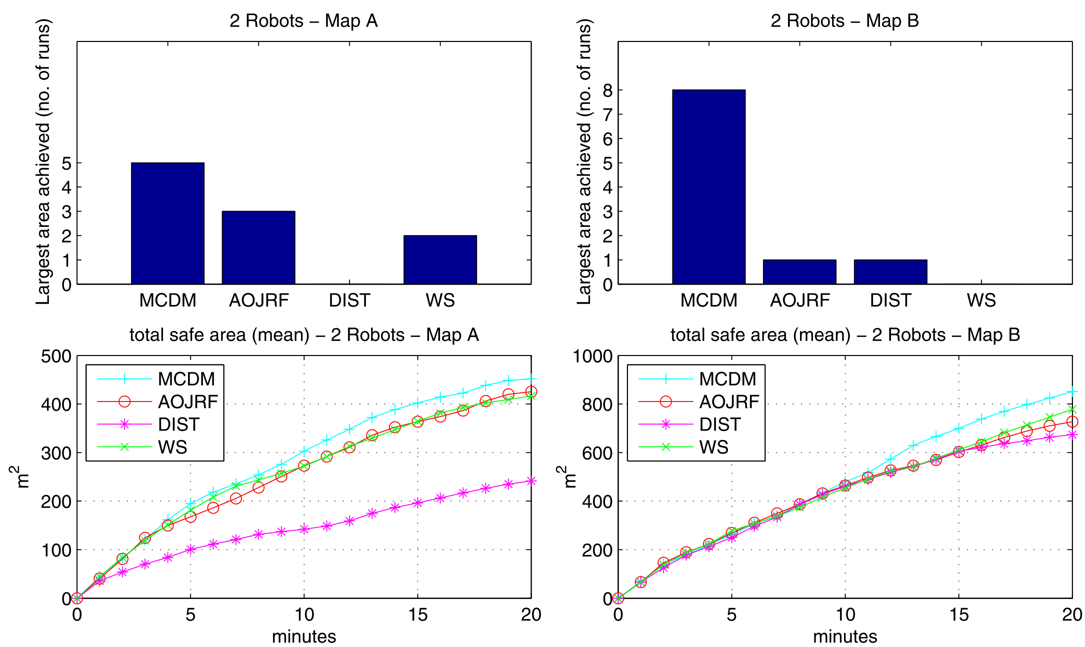
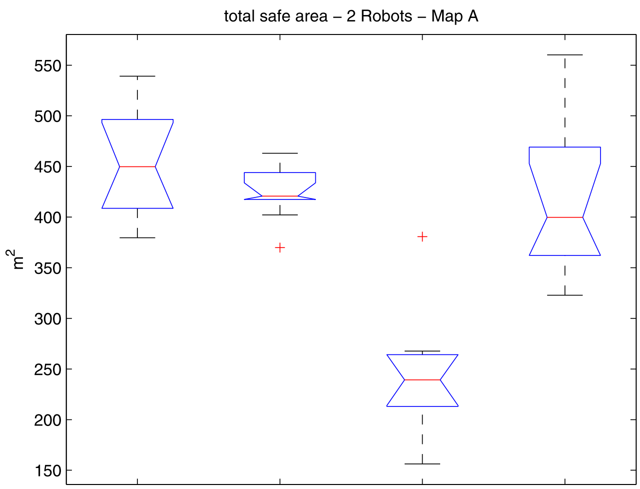
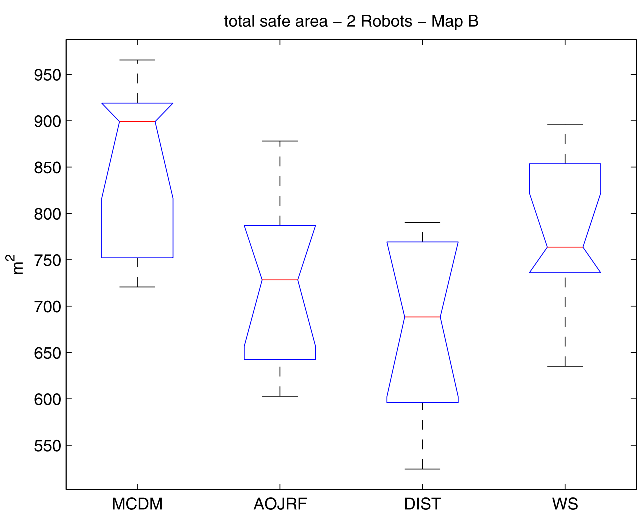


Fig. 4 Comparison between MCDM and other exploration strategies with two robots

similar in the contribution they can give to the explored area. Differently, in Map B the situation in which one alternative is remarkably better than others is more frequent. Consider, for instance, a frontier that lies close to an obstacle (from where an observation will return a small new area) and another one in front of an open space (from where an observation will return a large new area). In such situation, the benefits provided by a "right choice" would be more evident. This is what happens during the exploration of Map B, showing why differences between strategies are more statistically significant in this environment. This basically confirms results presented in Basilico and Amigoni (2009) for



(a)



(b)

Fig. 5 Statistical comparison of safe area mapped in 20 minutes

map building, reinforcing the idea that when very different alternatives are present and making a good choice is very rewarding, MCDM-based exploration strategies achieve satisfactory results.

Figure 6 shows the performance of the three MCDM-based exploration strategies with two robots (we omit results with one robot, for which the same considerations can be drawn). In both maps, MCDM, MCDMb, and MCDMw do not show statistically significant differences (p-values range from 0.0102, for MCDM and MCDMb in Map B, to 0.8652, for MCDMb and MCDMw in Map A). A first comparison that is worth doing is between MCDM and MCDMb strategies, to assess the impact of including criterion b in evaluating a location. Although their performance is very similar in exploring Map A, this performance is obtained moving the robots in rather different ways. The effects of introducing criterion b can be qualitatively noted by looking at the final maps built by the robots. A representative example is shown in Fig. 7, which reports the maps obtained with MCDM and MCDMb strategies after a run (with the same starting locations for the robots). Considering that the criterion b pushes the robots to discard locations that require complicate paths with several rotating maneuvers, the robots avoid to deeply explore corners, rooms, and other cluttered parts of the environment, preferring corridors and open spaces. The result is that the obtained map, from the one hand, is less precise but, from the other hand, is more representative of the general topology of the environment. This kind of map can be more useful to first responders (as discussed in Balaguer et al. 2009). The introduction of the criterion b does not show the same qualitative behavior in Map B, where the presence of open spaces makes intricate paths very rare. The use of this criterion in an open space is not justified by the characteristics of the environment, showing an example where "too much" informed local decisions could achieve a poor global performance.

Adopting different behaviors with the MCDMw strategy shows some (not statistically significant, see above) advantages in Map A. Roughly speaking, this strategy combines the benefits of MCDM and MCDMb strategies. In the first half of the exploration a more aggressive behavior is adopted, trying to maximize the explored area. Then, as the residual time decreases, the strategy becomes more conservative, avoiding cluttered zones and saving time. In Map B, the use of μ_1 in the first part of the exploration showed the main drawback of a very aggressive behavior: its vulnerability to decisions that happen to be not as good as expected. Qualitatively looking at the experiments, we noted that, in some situations, μ_1 pushed the robots to cover long distances for reaching locations with potentially large amounts of new area that, due to information gain estimation errors, were not so informative as expected once reached. This is the reason why MCDM and MCDMw curves are relatively separated in the first 10 minutes of exploration (Fig. 6).

Figure 8 depicts an example of paths followed by a robot in Map A, according to the three MCDM-based strategies. The starting location in all the three cases is in the center of the top corridor. All the strategies initially drive the robot toward the right part of the top corridor until the first difference can be observed in the path of MCDMw. Its aggressive behavior pushed the robot to go back at the intersection with the vertical corridor to obtain a wide view over the free space. MCDM and MCDMb start to differ in the bottom end of the central vertical corridor. More precisely, MCDMb path resulted more regular than that of MCDM. Indeed, MCDM drove the robot to explore a sequence of rooms while, with MCDMb, the robot chose to enter the bottom horizontal corridor. MCDMw's paths avoided all the rooms in the right part of the environment (first 10 minutes) but performed a more detailed exploration in the left

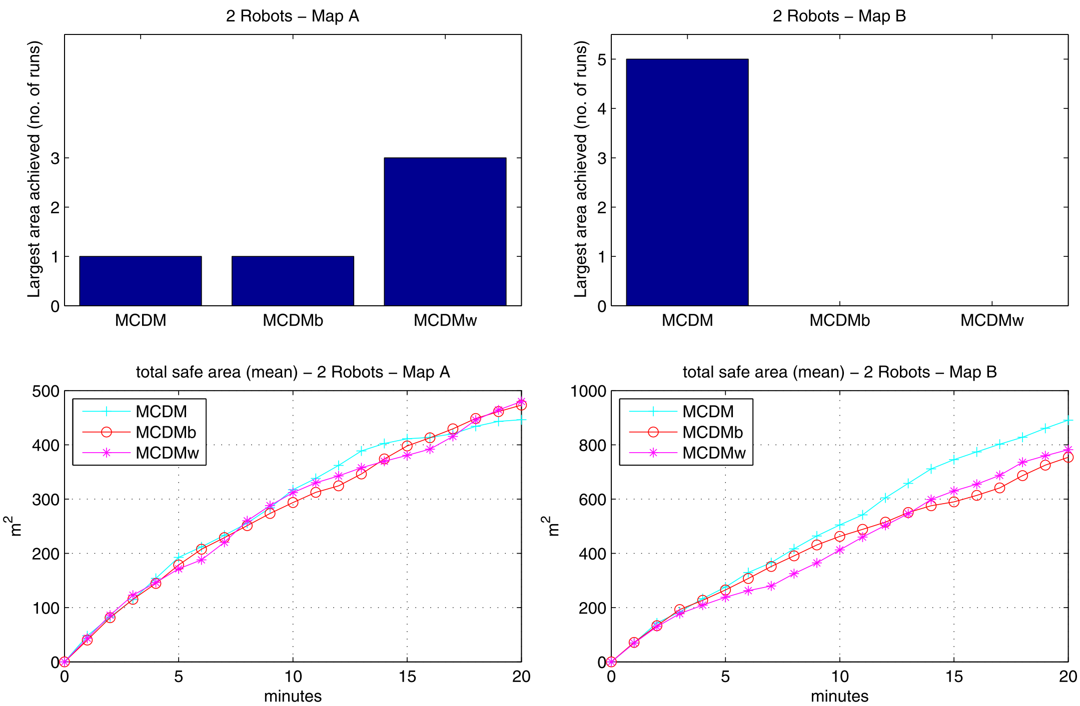
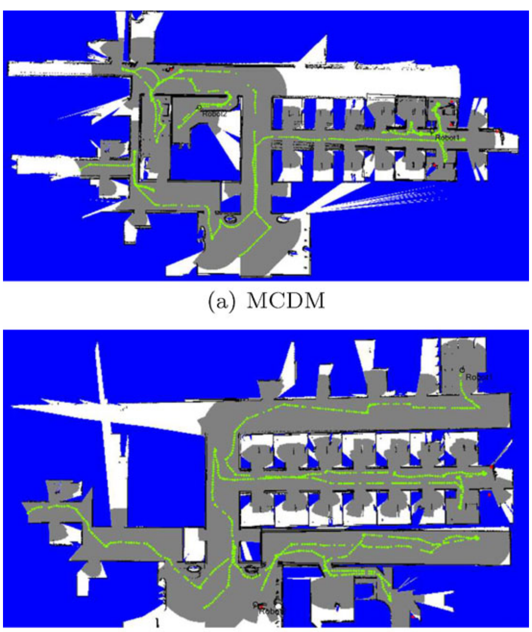


Fig. 6 Comparison between the three MCDM-based strategies



(a) MCDM

(b) MCDMb

Fig. 7 An example of maps obtained by two MCDM-based strategies

part of the map (last 10 minutes of the exploration). This example shows how obtained paths are coherent with the design principles of each strategy and demonstrates that the decision-theoretic framework provided by MCDM allows to define different exploration strategies whose behaviors exhibit some level of predictability.

We also evaluated the robustness of MCDM-based strategies with respect to small changes in the values of weights. We generated some sets of weights starting from the set reported in Table 1 (top) by randomly changing the weights of single criteria of $\pm 10\%$ and changing the weights of groups of criteria accordingly. For each set of weights, we performed some runs in the two maps with one or two robots, obtaining in each case a performance similar to that obtained with the original weights. From this set of experiments, we can conclude that MCDM-based exploration strategies are robust with respect to small variations in the values of the weights. Moreover, we evaluated the dependence of MCDM-based strategies on how candidate locations are selected on frontiers. Figure 9 shows the average performance of the MCDM strategy over 5 runs in Map B with one robot, according to two ways of selecting candidate locations on frontiers: as middle points (as otherwise done in this paper, according to Sect. 4.1) or as randomly selected locations on the frontiers. As it can be seen, the better performance of the selection of middle points becomes more evident as exploration proceeds. This results provides an *a posteriori* justification for the choice of considering candidate location as middle points of frontiers.

From our results, we can say that MCDM can be an effective method for defining good exploration strategies in search and rescue applications in initially unknown environments. Local decisions made with MCDM-based exploration strategies resulted in comparable, and sometimes better, performance when compared to other exploration strate-

Fig. 8 Paths followed by a robot using the three MCDM-based strategies

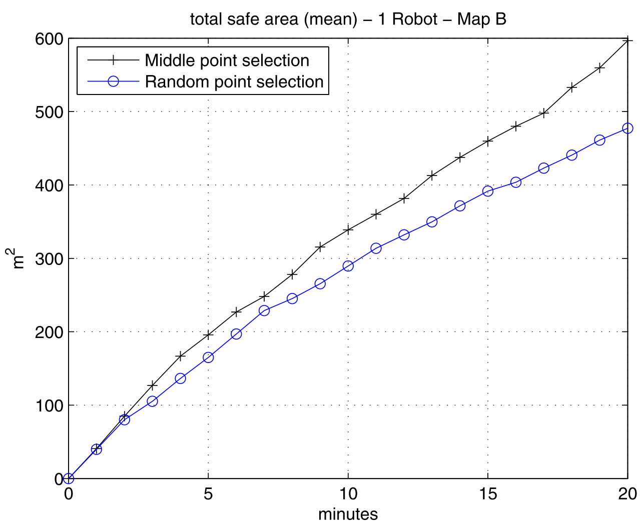


Fig. 9 Performance of the MCDM strategy with two different methods for selecting candidate locations

gies proposed in literature (e.g., in Map B, MCDM strategy performs significantly better than AOJRF strategy). In particular, MCDM-based strategies showed significant improvements in situations (like those faced in Map B) where making good decisions is more rewarding. Since MCDM is not an exploration strategy but a general method for defining exploration strategies and since our goal is not to find out the best exploration strategy, the fact that MCDM-based exploration strategies show performance comparable with that of other exploration strategies defined with *ad hoc* methods is satisfactory. Differently from *ad hoc* approaches, MCDM allows to define, by changing weights within the same formal framework, different exploration strategies that push the robots to exhibit rather different behaviors. In addition, MCDM presents a remarkable flexibility in composing criteria that can be exploited to add new criteria. In contrast, adding new criteria to *ad hoc* exploration strategies (like (3)) could be more complex.

6 Conclusions

In this paper, we have presented the application of the MCDM decision-theoretic approach to the definition of exploration strategies for searching initially unknown environments for fixed targets (victims) without any *a priori* information about their location. We have shown that MCDM provides a general and flexible way to develop utility functions for evaluating candidate observation locations in the context of an incremental exploration. Experimental results show that MCDM-based exploration strategies achieve a good performance, when compared with *ad hoc* strategies used in exploration with the advantage of a more flexible mechanism for composing criteria.

Possible future works include the development of adaptive methods that dynamically adjust the weights during exploration and the application of MCDM-based strategies to other search domains, like pursuit-evasion. Another interesting direction is working on the robot-frontier allocation methods, trying to achieve a closer integration between evaluation of candidate locations and coordination of robots. Finally, the practical utility of the proposed approach could be emphasized by developing a large catalogue of criteria from which designers can select those most suitable to address given applications. Introducing this catalogue would trigger a more extensive assessment of the potential of MCDM for defining exploration strategies.

Acknowledgements The authors would like to thank Alessandra Tonetti and Dario Facchi for their contributions to experimental activities.

References

Amigoni, F., & Caglioti, V. (2010). An information-based exploration strategy for environment mapping with mobile robots. *Robotics and Autonomous Systems*, *5*(58), 684–699.

Amigoni, F., & Gallo, A. (2005). A multi-objective exploration strategy for mobile robots. In *Proc. IEEE international conference on robotics and automation (ICRA)* (pp. 3861–3866).

Balaguer, B., Balakirsky, S., Carpin, S., & Visser, A. (2009). Evaluating maps produced by urban search and rescue robots: lessons learned from RoboCup. *Autonomous Robots*, *27*(4), 449–464.

Basilico, N., & Amigoni, F. (2009). Exploration strategies based on multi-criteria decision making for an autonomous mobile robot. In *Proc. European conference on mobile robotics (ECMR)* (pp. 259–264).

Burgard, W., Moors, M., & Schneider, F. (2005). Coordinated multi-robot exploration. *IEEE Transactions on Robotics*, *21*(3), 376–378.

Calisi, D., Farinelli, A., Iocchi, L., & Nardi, D. (2007). Multi-objective exploration and search for autonomous rescue robots. *Journal of Field Robotics*, *24*(8–9), 763–777.

Calisi, D., Iocchi, L., Nardi, D., Scalzo, C., & Ziparo, V. (2008). Contextual navigation and mapping for rescue robots. In *Proc. IEEE international symposium on safety, security, and rescue robotics (SSRR)* (pp. 19–24).

Carpin, S., Lewis, M., Wang, J., Balakirsky, S., & Scrapper, C. (2007). USARSim: a robot simulator for research and education. In *Proc. IEEE international conference on robotics and automation (ICRA)* (pp. 1400–1405).

Choset, H. (2001). Coverage for robotics: a survey of recent results. *Annals of Mathematics and Artificial Intelligence*, *31*(1–4), 113–126.

de Hoog, J., Cameron, S., & Visser, A. (2009). Role-based autonomous multi-robot exploration. In *Proc. computation world: future computing, service computation, cognitive, adaptive, content, patterns* (pp. 482–487).

Gerkey, B., & Mataric, M. (2004). A formal analysis and taxonomy of task allocation in multi-robot systems. *The International Journal of Robotics Research*, *23*, 939–954.

Gonzáles-Baños, H., & Latombe, J. C. (2002). Navigation strategies for exploring indoor environments. *The International Journal of Robotics Research*, *21*(10–11), 829–848.

Grabisch, M., & Labreuche, C. (2008). A decade of application of the Choquet and Sugeno integrals in multi-criteria decision aid. *4OR—A Quarterly Journal of Operations Research*, *6*(1), 1–44.

Grabisch, M., Kojadinovic, I., & Meyer, P. (2008) A review of capacity identification methods for Choquet integral based multi-attribute utility theory—applications of the Kappalab R package. *European Journal of Operational Research*, *186*(1), 766–785.

Hollinger, G., & Singh, S. (2010). Multi-robot coordination with periodic connectivity. In *Proc. IEEE international conference on robotics and automation (ICRA)* (pp. 4457–4462).

Khatib, O. (1986). Real-time obstacle avoidance for manipulators and mobile robots. *The International Journal of Robotics Research*, *5*(1), 90–98.

Kleiner, A., Prediger, J., & Nebel, B. (2006). RFID technology-based exploration and SLAM for search and rescue. In *Proc. IEEE/RSJ international conference on intelligent robots and systems (IROS)* (pp. 4054–4059).

Lin, L., & Goodrich, M. (2009). UAV intelligent path planning for wilderness search and rescue. In *Proc. IEEE/RSJ international conference on intelligent robots and systems (IROS)* (pp. 709–714).

Low, K., Dolan, J., & Khosla, P. (2008). Adaptive multi-robot wide-area exploration and mapping. In *Proc. international conference on autonomous agents and multiagent systems (AAMAS)* (pp. 23–30).

Marjovi, A., Nunes, J., Marques, L., & de Almeida, A. (2009). Multi-robot exploration and fire searching. In *Proc. IEEE/RSJ international conference on intelligent robots and systems (IROS)* (pp. 1929–1934).

Nevatia, Y., Stoyanov, T., Rathnam, R., Pfingsthorn, M., Markov, S., Ambrus, R., & Birk, A. (2008). Augmented autonomy: improving human-robot team performance in urban search and rescue. In *Proc. IEEE/RSJ international conference on intelligent robots and systems (IROS)* (pp. 2103–2108).

Pestman, W. (1998). *Mathematical statistics: an introduction*. Berlin: de Gruyter.

Rasche, C., Stern, C., Richert, W., Kleinjohann, L., & Kleinjohann, B. (2010). Combining autonomous exploration, goal-oriented coordination and task allocation in multi-UAV scenarios. In *Proc. international conference on autonomic and autonomous systems* (pp. 52–57).

Saeedi, P., Sorensen, S. A., & Hailes, S. (2009). Performance-aware exploration algorithm for search and rescue robots. In *Proc. IEEE international symposium on safety, security, and rescue robotics (SSRR)* (pp. 1–6).

Scone, S., & Phillips, I. (2010). Trade-off between exploration and reporting victim locations in USAR. In *Proc. IEEE international symposium on a world of wireless, mobile and multimedia networks (WoWMoM)* (pp. 1–6).

Singh, A., Krause, A., Guestrin, C., & Kaiser, W. J. (2009). Efficient informative sensing using multiple robots. *The Journal of Artificial Intelligence Research*, *34*(1), 707–755.

Stachniss, C., & Burgard, W. (2003). Exploring unknown environments with mobile robots using coverage maps. In *Proc. international joint conferences on artificial intelligence (IJCAI)* (pp. 1127–1134).

Tadokoro, S. (2010). *Rescue robotics*. Berlin: Springer.

Thrun, S. (2002). Robotic mapping: a survey. In *Exploring artificial intelligence in the New Millennium* (pp. 1–35).

Tovar, B., Munoz, L., Murrieta-Cid, R., Alencastre, M., Monroy, R., & Hutchinson, S. (2006). Planning exploration strategies for simultaneous localization and mapping. *Robotics and Autonomous Systems*, *54*(4), 314–331.

Visser, A., & Slamet, B. (2008). Including communication success in the estimation of information gain for multi-robot exploration. In *Proc. international symposium on modeling and optimization in mobile, ad hoc, and wireless networks (WiOPT)* (pp. 680–687).

Visser, A., de Buy Wenniger, G., Nijhuis, H., Alnajar, F., Huijten, B., van der Velden, M., Josemans, W., Terwijn, B., Sobolewski, R., Flynn, H., & de Hoog, J. (2009). Amsterdam Oxford joint rescue forces—team description paper—RoboCup 2009. In *Proc. RoboCup symposium*.

Wirth, S., & Pellenz, J. (2007). Exploration transform: a stable exploring algorithm for robots in rescue environments. In *Proc. IEEE international symposium on safety, security, and rescue robotics (SSRR)* (pp. 1–5).

Yamauchi, B. (1997). A frontier-based approach for autonomous exploration. In *Proc. IEEE international symposium on computational intelligence in robotics and automation (CIRA)* (pp. 146–151).

Nicola Basilico was born in Desio, Italy in 1981. He received a Master Degree in Computer Engineering at the Politecnico di Milano in July 2007. In February 2011 he received a Ph.D. in Computer Science from Politecnico di Milano with a work on navigation strategies for autonomous mobile robots employed in exploration and patrolling tasks. Since May 2011 he is a post-doctoral researcher at the University of California, Merced. His main research interests are navigation strategies for autonomous mobile robots, robotic patrolling, security games and algorithmic game theory. Other research interests include agents and multiagent systems.

Francesco Amigoni was born in Soresina (CR), Italy, on December 2, 1971. He got the Laurea degree in Computer Engineering from the Politecnico di Milano in 1996 and the Ph.D. degree in Computer Engineering and Automatica from the Politecnico di Milano in 2000. From December 1999 to September 2000 he has been a visiting scholar at the Computer Science Department of the Stanford University (USA). From February 2002 to April 2007 he has been an assistant professor at the Dipartimento di Elettronica e Informazione of the Politecnico di Milano. From May 2007 he is an associate professor at the Dipartimento di Elettronica e Informazione of the Politecnico di Milano. His main research interests include: agents and multiagent systems, mobile robotics, and the philosophical aspects of artificial intelligence. He is author of about 70 papers published on international journals, books, and conference proceedings.