Deception in Networks of Mobile Sensing Agents

Viliam Lisý[1], Roie Zivan[2], Katia Sycara[2]
and Michal Pěchouček[1]

[1]Agent Technology Center, Dept. of Cybernetics
FEE, Czech Technical University
Technická 2, 16627 Prague 6, Czech Republic
{lisy,pechoucek}@agents.felk.cvut.cz

[2]Robotics Institute, Carnegie Mellon University
5000 Forbes Avenue, Pittsburgh, PA, i5213, USA
{zivanr,katia}@cs.cmu.edu

ABSTRACT

Recent studies have investigated how a team of mobile sensors can cope with real world constraints, such as uncertainty in the reward functions, dynamically appearing and disappearing targets, technology failures end changes in the environment conditions.

In this study we consider an additional element, deception by an adversary, which is relevant in many (military) applications. The adversary is expected to use deception to prevent the sensor team from performing its tasks. We employ a game theoretic model to analyze the expected strategy of the adversary and find the best response. More specifically we consider that the adversary deceptively changes the importance that agents give to targets in the area. The opponent is expected to use camouflage in order to create confusion among the sensors regarding the importance of targets, and reduce the team's efficiency in target coverage. We represent a Mobile Sensor Team problem using the Distributed Constraint Optimization Problem (DCOP) framework. We propose an optimal method for the selection of a position of a single agent facing a deceptive adversary. This method serves as a heuristic for agents to select their position in a full scale problem with multiple agents in a large area. Our empirical study demonstrates the success of our model as compared with existing models in the presence of deceptions.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*multi-agent systems*

General Terms

Algorithms

Keywords

Deception, Sensor network, Game theory, Distributed problem solving

1. INTRODUCTION

Recent studies have investigated how a team (or a network) of mobile sensing agents can cope with various realistic elements of real world situations. One study [3] has considered uncertainty in the reward functions of agents for different alternative combinations of positions. Jain et. al. propose different ways to balance between the exploitation of the information agents acquire during search and the potential of exploring positions which were not visited yet. Another study [13], considered the different dynamic elements of such a problem including new targets, disappearing targets, technology failures of sensors and environment conditions which may reduce the quality of agent's reports. [13] proposed a model which captures these dynamic changes in the application and distributed algorithms for dynamic adjustment of the agents' deployment to the evolving state of the problem.

While some of the relevant applications in which mobile sensor networks (MSN) are expected to be used, as the Rescue scenario [6] or Maximizing Radio Signal [3], are "peaceful" applications in which agents are combating the forces of nature in order to solve the problem, other (military) applications include an adversary which is expected to make attempts to prevent the team from performing their task. For such applications a game theoretic model is appropriate, which can analyze the expected actions of the adversary and find the best response to it.

We consider a MSN that needs to "cover" targets, i.e. to allocate groups of sensors to monitor targets. Sensors are allocated as a function of a target's importance with more important targets getting higher value/number of sensors. In addition, we address the possibility of an adversary to use means for deception that will affect the importance that agents give to targets. Opponents in a military application are expected to use camouflage in order to decrease the importance that the MSN will give targets of high importance and on the other hand attempt to make insignificant targets appear to be of high importance. Such deception would cause the MSN to select a deployment in which precious resources are used for the surveillance of insignificant targets while the targets of high importance are not covered properly.

Following [13], our approach consists in finding an optimal strategy for the selection of a position of a single agent, and empirically test the success of this optimal strategy when used by agents in a team of agents with a common goal.

The proposed optimal strategy for a single agent considers multiple targets with various degrees of importance to which an adversary can increase or decrease the reflected importance. This effect on the reflected importance is bounded. This bound represents the limitations of camouflage for realistic targets (i.e., it is not realistic to consider that a large army base is disguised as a bush or a single car as a brigade).

Cite as: Deception in Networks of Mobile Sensing Agents, Viliam Lisý, Roie Zivan, Katia Sycara, Michal Pěchouček, *Proc. of 9th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2010)*, van der Hoek, Kaminka, Lespérance, Luck and Sen (eds.), May, 10–14, 2010, Toronto, Canada, pp. 1031-1038

We applied our local optimal method to problems with multiple agents in which agents have mobility and sensing limitations. In the distributed (local search) algorithm, agents share their positions with their neighbors and select their position using an algorithm based on the locally optimal method.

Our empirical study evaluates the success of both the single agent method and the local search algorithm for the agent team. In the case of the single agent problem, our proposed method is optimal in the worst scenario case, however, if the adversary is very limited or does not use its deception capabilities, a naïve method can produce high quality results. The same phenomenon was consistently found for the agent team. An experimental comparison of the proposed local search algorithm compared to a naïve local search algorithm, DSA reveals that the success of the proposed local search algorithm is more apparent when the bounds on the deception capabilities of the adversary are less tight.

The rest of this paper is organized as follows: Section 2 presents related work on deceptions in multi-agent applications. In Section 3 we introduce deceptions into the standard DCOP_MST model. Section 4 presents the optimal method for selecting the position of a single agent. Section 5 presents how the proposed method for a single agent is integrated in full case scenario. Our experimental evaluation is presented in Section 6 followed by our conclusions.

2. RELATED WORK

Deception has been studied in a limited way in the multi-agent literature. The work that is most similar to the presented research is [2]. The authors investigate deception in a simple two-player zero-sum game. One of the players is the attacker that decides to attack one of two targets. The defender can distribute 3 units to defend the targets. The more units are defending the attacked target, the lower is the reward for the attacker. Each of the defending units can be observed with one of two probabilities. The lower is the natural chance to observe the unit and the higher corresponds to the defender intentionally showing the defending unit in order to deceive the attacker. The authors analyze the optimal strategies for a player that by manipulating the information revealed to her opponent, she is rendering the observations of units useless. We perform a similar analysis for the case of mobile sensors, which requires a novel approach and optimizations.

Deception in the form of lowering the utility of the available information is investigated also in [8]. The task there is to plan paths for a team of UAVs, which inspect a given path to be used later by a convoy of ground vehicles, in such a way as to give an adversary that would want to ambush the convoy as little information as possible. Besides flying over the desired path, the UAVs would randomly fly over the other paths to deceive the observer.

Other analysis of deception in multi-agent systems include the control mechanism for a team of UAVs creating a single "phantom" aircraft by their movements [11] and the study [7] that suggests using software decoys in network security. These decoys should deceive the attacker to think the attack was successful, make him continue and allow assessing the nature of its attack.

Deception was studied formally also in the field of game theory. A formal deception game was first formulated as an open problem in [10]. One player is given a vector of three random numbers from uniform distribution on [0,1]. It changes one of the numbers to an arbitrary number from [0,1] and presents the modified vector to the second player. The second player chooses one position in the vector and receives as its reward the number that was originally on that position. The open question stated in the paper is whether there is a better strategy than randomly choosing one of the positions.

The game was solved in [4], showing that for the case of 3 numbers, the information provided to second player can be made completely useless, but in case of 4 numbers and changing only one of them, the expected gain of the optimal strategy of the second player is more than the mean value guaranteed by random choice.

A generalized form of the game is solved in [1]. The game is played with vector of arbitrary length (n). The first player is allowed to *permute* the vector in a way that only up to m numbers change their positions. The second player then selects the position and obtains the reward corresponding to the number at the position in the original vector. The paper shows that if the first player is allowed to change at least half of the positions, the resulting vector would not contain any information useful for selecting a high number by player two.

The model presented in this paper differs from the previous models by allowing the first player to modify each of the numbers, but only by a given amount. This assumption leads to novel method for creating the selection strategy that is not based on the results from the papers above.

3. PROBLEM DEFINITION

Our problem definition is as in [13] with addition of possible deception. The task is placing a network of mobile sensing agents to suitable positions, so that they meet the specified requirements for surveillance of individual targets, i.e. points in space. The problem is set to a discretized metric space with a finite set of positions. A network of finite number of agents $A_1, \ldots, A_n$ operates in the space. Each of the agents (A_i) is placed in some position cur_pos_i and it is characterized by three parameters. The *sensing range* (SR_i) is the effective coverage range of the agent, i.e., agent A_i can detect and cover all the targets that are within its sensing range from cur_pos_i. The *mobility range* (MR_i) is the range that an agent can move in a single time step. And the *credibility* $Cred_i$ is a real positive number representing the quality of the sensor.

We further define an environmental requirement function ER, i.e. importance of covering a position. This function expresses for each point in the area, the required joint credibility amount (the sum of the credibility variables) of agents that have this point within their sensing range (i.e. covered). Function Cur_DIFF calculates for each point in the area the difference between the current value of the ER function and the sum of the credibilities of the agents which are currently covering it. Formally, if we denote the set of agents within their sensing ranges from point p by SR_p then:

$$Cur\_DIFF(p) = ER(p) - \sum_{A_i \in SR_p} Cred_i \qquad (1)$$

The global goal of the agents is to cover all the targets according to ER (i.e. to reduce the largest value of Cur_DIFF to zero) in a pre-defined number of time steps. Since this goal cannot always be achieved, we define a more general goal which is to minimize the largest value of the Cur_DIFF function over all targets in the area.

In addition to these original problem properties, we define the properties relevant for the deception. We assume that

the adversary with limited capabilities has used means of deception to make the environmental requirements appear to be $v(p)$ for each position p. The sensors do not have the knowledge about the *real* importance of the target ($ER(p)$), but only about the *apparent* importance $v(p)$. Moreover, the sensors have some information about the adversary capabilities, options and effort put into deception. This knowledge is represented by a pair of functions Δ^+ and Δ^-. These functions relate the real requirements to the apparent ones for the individual points in space. For each point in space, the unknown real importance of the target ($ER(p)$) can be any number between $v(p) - \Delta^+(p)$ and $v(p) + \Delta^-(p)$, i.e. Δ^+ and Δ^- are the maximal increment and decrease, respectively, that the adversary can cause to any target.

The goal of the sensors remain to cover the *real* requirements given by $ER(p)$ for the individual positions.

4. FORMAL DECEPTION GAME

This section presents one of the most important contributions of the paper. It shows the derivation of the sensor placement method that is robust to deception. It is a probabilistic method that allows choosing the targets with high importance while keeping the right amount of randomization to prevent the adversary to mislead via deception.

4.1 Formal Game Definition

We define the problem being solved in this section as a game between a single sensor and an adversary that uses deception (such as decoys or camouflage) to make the sensor less efficient. The adversary starts with n targets of various positive importance values $(y_1, \ldots, y_n)$. It can use deception in order to generate a perceived importance of each target $(v_1, \ldots, v_n)$, but the types of the targets and the effort he can spend on the camouflage for each of the targets does not allow him to modify the importance of the target arbitrarily. The perceived importance for each target is bounded to an interval:

$$v_i \in [y_i - \Delta_i^-, y_i + \Delta_i^+]$$

where Δ_i^+ as well as Δ_i^- are non-negative real numbers and $y_i - \Delta_i^- \geq 0$ for each target i. The sensor can perceive the modified importance values and we assume it knows the boundaries Δ_i^+, Δ_i^-. Based on this information, it decides to cover a single target, trying to maximize its perception of the target's real importance. We represent the strategy of the sensor as a probability distribution of covering individual targets $(x_1, \ldots, x_n)$.

The goal of this paper is to create a sensor placement method that would be robust against deception, so we perform analysis of the worst case scenario. The natural formalization of the problem would then be maximizing the expected value of the covered target in case of the worst case $\vec{y}$ that is consistent with the observations $\vec{v}$.

$$\max_{\vec{x}} \min_{\vec{y}} \sum_{i=1}^{n} x_i y_i$$

However, this optimization is trivial. For any fixed $\vec{x}$, the worst case corresponds to setting all $y_i = v_i - \Delta_i^-$. It says that the sensor covers less important targets in case that all the targets are generally less important. It constitutes the worst case in problem instance rather than some smart information manipulation by the adversary. In order to focus on adversary deceptive strategies, we use a simple observation.

If we assume that the sensor player is rational (makes optimal decisions), it should be able to figure out when the information provided by the adversary is useless and hence it should always reach at least the payoff of random strategy corresponding to uniform probability distribution over the targets.

$$\frac{1}{n} \sum_{i=1}^{n} y_i$$

We can reformulate the objectives of the players relative to this assured value. The sensor player tries to gain more than what is provided by the random strategy and the adversary uses deception to make the information useless and force the opponent to play the random strategy. Formally, the task that the sensor solves is

$$\begin{array}{ll} \max_{\vec{x}} \min_{\vec{y}} \frac{\sum_{i=1}^{n} x_i y_i}{\frac{1}{n}\sum_{j=1}^{n} y_j} & \text{s.t.} \\ & \vec{0} \leq \vec{x} \leq \vec{1} \\ & \sum_{i=1}^{n} x_i = 1 \\ & \vec{v} - \vec{\Delta^+} \leq \vec{y} \leq \vec{v} + \vec{\Delta^-} \end{array} \quad (2)$$

This problem formulation still requires finding the strategy that assures the highest expected coverage, but it removes the trivial cases that prevent more interesting solutions of the game.

4.2 Worst Case for Fixed Sensor Strategy

In this subsection, we show what is the worst case $\vec{y}$ for any fixed $\vec{x}$.

LEMMA 4.1. *The function* $f(\vec{x}, \vec{y}) = \frac{\sum_{i=1}^{n} x_i y_i}{\frac{1}{n}\sum_{j=1}^{n} y_j}$ *is monotonic in any* y_c *if all other* y_i *and* x_i *are fixed.*

PROOF. For arbitrary $c \in 1..n$, we compute the sign of the partial derivative with respect to y_c. The constant in the denominator can be omitted.

$$\operatorname{sgn}\left(\frac{\partial}{\partial y_c}\left(\frac{\sum_{i=1}^{n} x_i y_i}{\sum_{j=1}^{n} y_j}\right)\right) = \operatorname{sgn}\left(\frac{x_c \sum_{j=1}^{n} y_j - \left(\sum_{i=1}^{n} x_i y_i\right) 1}{\left(\sum_{j=1}^{n} y_j\right)^2}\right) = \operatorname{sgn}\left(\sum_{i=1}^{n} (x_c - x_i) y_i\right) \quad (3)$$

In case of $i = c$ the term $(x_c - x_i) = 0$ and it makes formula (3) independent of the value of y_c. The sign of the derivative is constant in y_c, hence the function is monotonic in y_c. □

An immediate corollary of the fact above is that in finding the worst case $\vec{y}$, we need to consider only the vectors with all their coordinates set to the extreme values.

COROLLARY 4.2. *For all the coordinates in the worst case* $\vec{y}$, $y_i = v_i - \Delta_i^+$, $y_i = v_i + \Delta_i^-$ *or the coordinate does not influence the optimized value.*

PROOF. Let $\vec{y}$ be the worst case for a fixed $\vec{x}$ and consider an arbitrary coordinate y_i. Lemma 4.1 says that the sign of the derivative of f according to y_i is constant. If it is (constantly) zero, the optimized value does not depend on the value assigned to y_i. Otherwise f is either strictly increasing or strictly decreasing in y_i. If the sign of the derivative is (constantly) $+1$, any $y_i > v_i - \Delta_i^+$ can be decreased to $y_i = v_i - \Delta_i^+$ decreasing the optimized function. If it is (constantly) -1, any $y_i < v_i + \Delta_i^-$ can be increased to $y_i = v_i + \Delta_i^-$ decreasing the optimized function. □

LEMMA 4.3. *For any fixed $\vec{x}$, there exists a coordinate $c \in 1 \ldots n$, such that the worst case $\vec{y}$ can be constructed as*

$$\begin{array}{ll} \text{if } x_i \geq x_c & \text{then } y_i = v_i - \Delta_i^+ \\ & \text{else } y_i = v_i + \Delta_i^- \end{array}$$

PROOF. The proof is by induction on the number of unset coordinates. First we show that we can always set some coordinates of $\vec{y}$ in the global optimum at the beginning. Then we show that after setting any number of coordinates, we can find one more that can be set in the global optimum.

I) Assume the coordinates

$$b = \arg\min_{i \in 1..n} x_i,\ t = \arg\max_{i \in 1..n} x_i$$

Then for any $i \in 1..n$ holds

$$((x_b - x_i)y_i) \leq 0,\ ((x_t - x_i)y_i) \geq 0$$

As a result, for the sign of the partial derivative in formula (3) the following holds:

$$\text{sgn}\left(\sum_{i=1}^{n}(x_b - x_i)y_i\right) \leq 0,\ \text{sgn}\left(\sum_{i=1}^{n}(x_t - x_i)y_i\right) \geq 0$$

for all possible values of other x_i and y_i. That means that in the globally optimal $\vec{y}$, we can set

$$y_b = v_b + \Delta_b^-,\ y_t = v_t - \Delta_t^+$$

II) Assume that some of the values y_i are already set. Without loss of generality rename the coordinates so that the set coordinates are $1, \ldots, k$. Then

$$S_c = \text{sgn}\left(\underbrace{\sum_{i=1}^{k}(x_c - x_i)y_i}_{A_c} + \sum_{i=k+1}^{n}(x_c - x_i)y_i\right)$$

and the term A_c is a fixed constant for each c. Using the same argument as in part I)

(a) if $c_t = \arg\max_{i \in k+1..n} x_i$ & $A_{c_t} \geq 0$ then $S_{c_t} \geq 0$ for all x_i, y_i and we can set $y_{c_t} = v_{c_t} - \Delta_{c_t}^+$

(b) if $c_b = \arg\min_{i \in k+1..n} x_i$ & $A_{c_b} \leq 0$ then $S_{c_b} \leq 0$ for all x_i, y_i and we can set $y_{c_b} = v_{c_b} + \Delta_{c_b}^-$

In order to finish the proof, we have to show that at least one of the conditions in (a),(b) always holds. Clearly $x_{c_t} \geq x_{c_b}$. Assume that (b) does not hold

$$A_{c_b} > 0 \Leftrightarrow \sum_{i=1}^{k}(x_{c_b} - x_i)y_i > 0 \xrightarrow{x_{c_t} \geq x_{c_b}}$$
$$\sum_{i=1}^{k}(x_{c_t} - x_i)y_i > 0 \Leftrightarrow A_{c_t} > 0 \Rightarrow (a) \text{ holds}$$

The induction always sets the y_i for the unset target with minimal x_i to $v_i + \Delta_i^-$ or the y_i corresponding to the unset target with maximal x_i to $v_i - \Delta_i^+$. As a result, if all of the coordinates are set, the last set coordinates define the x_c from the proposition. □

4.3 Optimal Strategy for the Sensor

In this section, we show how the optimal solution for the sensor in the deception game can be computed by a linear program. The linear program is based on the standard trick often used in game theory. The *max-min* optimization from formula (2) can be rewritten to *max* optimization for the price of defining additional constraints (e.g. [9]). A constraint must be added for each strategy $\vec{y}$ that can possibly be the worst case for a valid strategy of the maximizing player ($\vec{x}$). Corollary 4.2 assures that we need to add just a finite number of constraints.

COROLLARY 4.4. *If we denote the set of all targets T, the optimal strategy for the sensor in the deception game can be computed by the linear program*

$$\begin{array}{ll} \max_{\vec{x},z}\ z & s.t. \\ & \vec{1} \geq \vec{x} \geq \vec{0} \\ & \sum_{i=1}^{n} x_i = 1 \\ & \forall A \subseteq T\ \frac{\sum_{i \in A} x_i(v_i - \Delta_i^+) + \sum_{i \in (T \setminus A)} x_i(v_i + \Delta_i^-)}{\sum_{i \in A}(v_i - \Delta_i^+) + \sum_{i \in (T \setminus A)}(v_i + \Delta_i^-)} \geq z \end{array} \quad (4)$$

If we use this basic program, the number of constraints is exponential in the number of targets. The rest of this section shows, how the number of constraints can be further reduced. First, we need to define the notion of partial ordering of the targets and present a technical lemma.

DEFINITION 4.5. *We define the relation $t_i \trianglerighteq t_j$ on the targets that have both the upper and lower bounds of their intervals ordered.*

$$(v_i - \Delta_i^+) \geq (v_j - \Delta_j^+)\ \&\ (v_i + \Delta_i^-) \geq (v_j + \Delta_j^-)$$

Note, that this relation is transitive, because of the transitivity of the ordering of the bounds.

LEMMA 4.6. *If $t_k \trianglerighteq t_l$ then there is an optimal strategy for the sensor for which $x_k \geq x_l$.*

PROOF. Remember that the worst case for any sensor strategy is modifying the importance of the targets to the bound of the interval. If both of the inequalities in definition of $\trianglerighteq$ hold as equalities, the ordering of the probability of covering them (x_k, x_l) cannot make difference for any of the players.

Next we assume both inequalities defining $t_k \trianglerighteq t_l$ are strict. We consider a modified game, where the players are presented the real importance of the targets and they simultaneously decide on which target to cover on the sensor side and how to modify the real importance of the targets on the adversary side. The optimal strategy for the sensor in this game corresponds to the deception-robust play in the original game. The *max-min* oprimization for both games is formula (2). The game is a two player zero-sum game with finite number of strategies on each side and hence it has a Nash equilibrium $(\vec{x}, \vec{y})$ with a unique value (e.g. [9]).

With coordinates renamed such that $\forall i \in 2..n\ x_{i-1} \geq x_i$, Lemma 4.3 says that there is a coordinate c, such that $\vec{x}$ optimizes

$$\frac{n}{\sum_{j=1}^{n} y_j}\left(\sum_{i=1}^{c-1} x_i(v_i - \Delta_i^+) + \sum_{i=c}^{n} x_i(v_i + \Delta_i^-)\right)$$

We show that if $x_k < x_l$ then swapping values of x_k and x_l improves the strategy of the sensor player for the fixed $\vec{y}$ which would be a contradiction with $(\vec{x}, \vec{y})$ being the Nash equilibrium. The fixed strategy of the adversary implies that the denominator of the optimized function stays the same.

For $l < c \leq k$ the strategy is improved if

$$x_l(v_l - \Delta_l^+) + x_k(v_k + \Delta_k^-) < x_k(v_l - \Delta_l^+) + x_l(v_k + \Delta_k^-)$$
$$(x_l - x_k)(v_l - \Delta_l^+) + (x_k - x_l)(v_k + \Delta_k^-) < 0$$
$$\underbrace{(x_l - x_k)}_{>0}\left[(v_l - \Delta_l^+) - (v_k + \Delta_k^-)\right] < 0$$
$$(v_l - \Delta_l^+) < (v_k + \Delta_k^-)$$

The last inequality holds from $t_k \trianglerighteq t_l$, because

$$(v_l - \Delta_l^+) \leq (v_l + \Delta_l^-) < (v_k + \Delta_k^+)$$

For $l < k < c$ the strategy is improved if

$$x_l(v_l - \Delta_l^+) + x_k(v_k - \Delta_k^+) < x_k(v_l - \Delta_l^+) + x_l(v_k - \Delta_k^+)$$
$$(v_l - \Delta_l^+) < (v_k - \Delta_k^+)$$

For $c \leq l < k$ the strategy is improved if

$$x_l(v_l + \Delta_l^-) + x_k(v_k + \Delta_k^-) < x_k(v_l + \Delta_l^-) + x_l(v_k + \Delta_k^-)$$
$$(v_l + \Delta_k^+) < (v_k + \Delta_l^-)$$

All the final inequalities trivially hold from $t_k \trianglerighteq t_l$.

The last two cases to finish the proof are if just one of the inequalities holds strictly. If $(v_k - \Delta_k^+) = (v_l - \Delta_l^+)$ and $(v_k + \Delta_k^-) > (v_l + \Delta_l^-)$ then for cases besides $l < k < c$ the argument about improving the value for the sensor player above holds. In this case, the pay-off of the sensor player does not change, but we show that if the sensor player switches the probabilities of covering targets t_k and t_l, the adversary does not have an incentive to deviate from its original strategy. This means that the original strategy of the adversary and the modified strategy of the sensors form a Nash equilibrium.

Remember the construction of the worst response of the adversary in the proof of Lemma 4.3. If the adversary modifies the real importance of the targets one by one, it arrives first to x_l as before. All the targets that are camouflaged so far are set on the same values as for the unmodified sensor strategy. The direction in which the target t_k will be camouflaged does not depend on the bounds on y_k. It depends only on x_l and the already camouflaged targets. As a result the target t_k, currently corresponding to the value x_l will be camouflaged in the same direction as t_l before, setting it to the same value. If construction of the optimal strategy for the adversary continues and encounters x_k, all the data used to compute the direction of deceptively altering the target t_l are the same as with the original sensor strategy and it is altered to the same value as t_k before. That is why the worst response to the modified sensor strategy is the same as to the original sensor strategy. The proof of the last case where the second condition holds as equality is symmetric. □

Using the previous lemma, we can reduce the number of constraints needed in the linear program (4).

THEOREM 4.7. *The optimal solution for the sensor player can be computed by the linear program*

$$\begin{array}{ll} \max_{\vec{x},z} z & s.t. \\ & \vec{1} \geq \vec{x} \geq \vec{0} \\ & \sum_{i=1}^{n} x_i = 1 \\ & \forall t_i \trianglerighteq t_j, \nexists t_k\ t_i \trianglerighteq t_k \trianglerighteq t_j \\ & \quad x_i \geq x_j \\ & \forall A \in T : \nexists i \in A, j \in (T \setminus A) t_j \trianglerighteq t_i \\ & \quad \frac{\sum_{i \in A} x_i(v_i - \Delta) + \sum_{i \in (T \setminus A)} x_i(v_i + \Delta)}{\sum_{i \in A}(v_i - \Delta) + \sum_{i \in (T \setminus A)}(v_i + \Delta)} \geq z \end{array} \quad (5)$$

PROOF. From Lemma 4.6, we know that introducing the constraints on the ordering of x_i will not prevent us from finding the optimal solution. The linear program finds the correct solution, if for any strategy $\vec{x}$ that respects the ordering, a constraint representing the worst response to the strategy is present. From Lemma 4.3, we do not have to consider that the adversary decreases the importance of a target that is covered with higher probability than another target for which it increased the importance. □

Now consider a basic setting, in which a single sensor can cover one of n targets and the sensor knows that the importance of each of the targets could have been modified by a fixed Δ up or down. Even if the resulting apparent importance is bounded to a predefined interval [0,MAX_IMP], the relation $\trianglerighteq$ creates a full ordering. It is the same ordering as the ordering of the apparent importance. For this kind of settings, the linear program computing the strategy for the sensor has only small number of constraints that is linear in the number of targets.

COROLLARY 4.8. *If the ordering induced by the relation $\trianglerighteq$ is full and if we rename the targets so that*

$$\forall i \in 2..n\ \ t_{i-1} \trianglerighteq t_i$$

then the optimal strategy for the sensor can be computed by the linear program

$$\begin{array}{ll} \max_{\vec{x},z} z & s.t. \\ & \forall j\ \frac{\sum_{i=1}^{j} x_i(v_i - \Delta_i^+) + \sum_{i=j+1}^{n} x_i(v_i + \Delta_i^-)}{\sum_{i=1}^{j}(v_i - \Delta_i^+) + \sum_{i=j+1}^{n}(v_i + \Delta_i^-)} \geq z \\ & \vec{1} \geq \vec{x} \geq \vec{0} \\ & \forall i \in 2..n\ \ x_{i-1} \geq x_i \\ & \sum_{i=1}^{n} x_i = 1 \end{array} \quad (6)$$

5. SENSOR NETWORK ALGORITHM

In this section, we present how finding the solution for the formal deception game described in the previous section can be used to create a successful deception-robust heuristic algorithm for the whole network. The first step of the generalization is constructing a local strategy for single agent with specified sensor and mobility ranges. This strategy can then be used as the local method for solving the whole problem in the DCOP framework via local search.

5.1 Sensors with Mobility and Sensing Range

The formal analysis in the previous section gives us a solution in case of covering just one target, i.e. the sensing range of the sensor is very small. It also assumes that the mobility restrictions of the sensors allow covering any of the targets considered. In order to loosen these restrictions, we design the local algorithm that choses the best position among all the positions in the mobility range of a single sensor. Considering the main criterion of minimizing the remaining importance of the most important uncovered target (see Section 3), the best position should cover the most important target and if the sensing range allows it, also the second, third and next most important targets. The local algorithm that does not take deception into account is described in [13]. Figure 1 presents its deception-aware variant. Each agent calls the described function with all targets and all positions in its sensor range.

The algorithm incrementally constructs the set of most important targets that is eventually covered by the sensor. The main issue is that with deception, we often cannot be sure which target is the most important. That is why the

Require: *posSet*: available positions for the sensor, *targets*: targets to consider, Δ: assumed capabilities of the adversary
Ensure: position where to move the sensor
1: $close := \{t \in targets : \exists p \in posSet\ dist(p,t) < SR\}$
2: **if** *close* is empty **then**
3: **return** Any $pos \in posSet$
4: **else if** $||close|| = 1$ **then**
5: $top :=$the only element of *close*
6: **else**
7: $top :=$ **deceptionAwareMax**$(close, \Delta)$
8: **end if**
9: $posSet := \{p \in posSet : dist(p, top) < SR\}$
10: **return** selectPosDA$(posSet, close \setminus top, \Delta)$

Figure 1: selectPosDA$(posSet, targets, \Delta)$ – local deception-aware procedure for selecting the best position for placing single sensor covering targets in circular sensing range (SR).

selection of the most important target from the remaining options (line 7) is performed by the deception-aware method developed in Section 4 as the procedure deceptionAwareMax. This procedure returns a single target that is selected according to the probability distribution resulting from the linear program in Theorem 4.7.

5.2 Multiple Coordinated Sensors

The generalization to the case of multiple coordinated sensors is performed using the DCOP_MST model introduced in [13]. This model, with some small adjustments, can be solved by standard DCOP local algorithms and the results in [13] show that a simple hill-climbing algorithm as MGM [5] or DSA [12] can be efficient in solving this problem without considering deception

The algorithm which we have chosen to incorporate our method within is the DSA algorithm. In DSA, agents share with their neighbors only their current assignments (position in our case). In each iteration of the algorithm, an agent finds the best alternative for her current assignment according to the assignments it received from her neighbors. It replaces her assignment by the alternative with a pre-defined probability p or retains her original assignment. In the MGM algorithm, agents share besides their assignments, the maximal improvement (reduction) in cost they can achieve by replacing their assignment. DSA was favored over the variants of the MGM since MGM requires that agents are able to exactly quantify the quality of this proposed local reductions, i.e. quantify the difference between the coverage of the importance in the current and the alternative positions. Using the deception aware algorithm, many local reductions have uncomparable quality. It is often not possible to say that one position is strictly better then another. The benefit of using the proposed algorithm demonstrates only statistically, if the agents follow the probability distribution given by the method.

If the apparent importance of the most important target covered form the alternative position reduced to the lower bound of the interval of its possible real importances is lower then then the apparent importance of the most important target covered from the current position increased to the upper bound of the interval of its possible real importances then the alternative position clearly dominates the current one. However if this does not hold, any ordering of the real importance of the positions is possible.

In order to allow similar grounds for the deception-aware

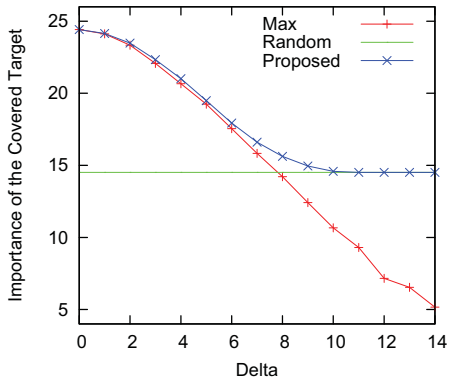

Figure 2: The importance of the target covered by the proposed and two baseline algorithms in formal deception game scenario. The adversary uses its capabilities optimally against the naïve approach.

and the naïve (deception ignoring) algorithm, we use a variant of DSA proposed by [12], that allows the sensors in the deception aware as well as deception ignoring variant to move not only in cases of positive local reduction, but also if the best new position is not worse than the current one. If we disallowed moving to the uncomparable states, the deception-aware algorithm would have very restricted exploration capablities.

6. EXPERIMENTAL EVALUATION

In this section, we experimentally evaluate the derived solution of the formal deception game as well as the performance of the proposed heuristic algorithm based on this method (presented in Figure 1).

6.1 Formal Deception Game

The first experiment we present validates the strategy formally derived in Section 4. The sensor has to choose to cover one of five targets with importance uniformly randomly selected from the interval [0,29]. The graph in Figure 2 shows the mean importance of the covered target from 1000 runs of the experiment and three target selection strategies. The horizontal axes of the graph represent the capability of the adversary (Δ), that is the same for all the targets and positive as well as negative modification of the importance.

$$\forall i, j \Delta_i^+ = \Delta_i^- = \Delta_j^+ = \Delta$$

In this case, the algorithm that the adversary uses to deceive the sensor is (a) finding the least important target t that can be camouflaged to be the most important one, i.e. such target that the importance of the most important target lowered by Δ is smaller than the importance of the target increased by Δ. (b) Increasing the importance of target t as well as all the targets with smaller importance by Δ and decreasing the importance of all targets that are more importance than t by Δ. This way, target t appears to be the most important target after the camouflage. The higher is the capability of the adversary, the higher is the chance that a target with low importance will appear to be the most important target.

Figure 2 shows the results for three target selection strategies. The naïve algorithm (denoted as *Max*) does not assume deception and selects the target that appears to be the most important. The real importance of the target chosen by this algorithm decreases with increasing capability of the adver-

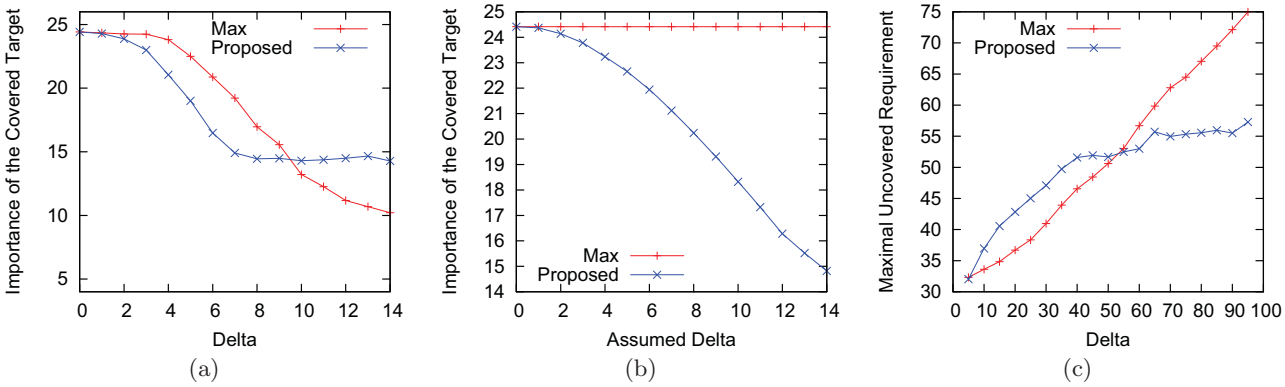

Figure 3: (a) The importance of the target covered by the proposed and two baseline algorithms on formal deception game scenario. The adversary uses sub-optimal heuristics. (b) The price of paranoia for the proposed algorithm on formal deception game scenario. (c) The quality of DSA algorithm on the multi-agent scenario with the proposed deception-aware and the apparent importance maximizing local method.

sary. If the adversary is able to modify the importance by more than half of the maximal importance of the targets, it can make the least important target appear to be the most important and as a result, the naïve algorithm selects the least important target.

The second algorithm is called *Random* in the figure. It completely ignores any information about the importance of the targets and selects one of them randomly. This leads to constant performance, corresponding to the mean real importance of the targets, independent on the adversary capabilities.

The third algorithm is the *Proposed* algorithm, selecting the target according to the distribution resulting from the linear program in Corollary 4.8, which is sifficient for this setting. In case the adversary is using its capabilities in the optimal way, the performance of the proposed algorithm is never worse than the performance of the Max algorithm nor the blind Random algorithm. This shows that the algorithm is capable of using the information that is still left in the importance value due to limited capabilities of the adversary, but it is robust against deception and if no useful information is present anymore, it chooses the target randomly.

6.1.1 *Sub-optimal Adversary*

The proposed algorithm was developed as a response to the worst possible real importance of the targets that is consistent with the observation. If the adversary does not use its capability optimally, additional information is left in the apparent importance and the proposed algorithm can perform worse than Max. This situation is shown in Figure 3(a). The setting is the same as in the previous case, but the adversary is using a simpler strategy. It (a) computes the mean importance of the targets in the current scenario and (b) increases the importance of all the targets with importance lower than the mean by Δ while increasing the importance of all the remaining targets by Δ.

This strategy of the adversary is far from the optimal strategy and it makes the proposed algorithm too conservative, randomizing even for the cases that still contain enough information about the real importance of the targets. However, even for this sub-optimal adversary, the proposed algorithm outperforms Max algorithm if the adversary is able to modify the importance more than one third of the maximal possible importance.

6.1.2 *Price of Paranoia*

We show above that the proposed method can be outperformed if the opponent does not use its capabilities optimally. Now we examine the decrease in the quality of the solution in case that no adversary is trying to deceive the sensor, but the sensor expects otherwise. We call the function, mapping the expected adversary capability to the decrease of the quality of produced solution (compared to the optimal solution without deception), *the price of paranoia*. For our scenario, the price of paranoia can be seen on Figure 3(b). The optimal solution without any deception is the Max algorithm producing the straight line in the figure. With increasing paranoia (i.e. the false belief about the adversary activity), the performance of the proposed algorithm gradually decreases to the mean real importance.

6.2 Full Scale Scenario

We continue by showing that the presented deception aware method can be successfully used as the method executed by each agent in the decentralized local search algorithm for solving the sensor placement problem defined in Section 3. We evaluate it on a grid of 50x50 positions with ten targets with random importance selected from interval [0,99] with uniform probability. The adversary uses a sophisticated heuristic method to camouflage the targets. The method is based on repeated selecting of a group of targets close to a very important target and making it look more important than the target. This group is selected as the least important group that can be camouflaged to be more important than the selected target and at the same time, the group cannot be covered together with the very important target. While making the camouflage, the adversary does not necessarily modify the targets as much as possible, only enough to make them look more important than the target it wants to draw the sensors' attention from (the real important target).

Ten identical agents are present in the scenario. The sensor range of all the agents is set to ten, the mobility range is set to twenty. The credibility of a sensor is thirty. Consequently, at least four sensors are needed to completely cover a target of full importance.

We use the same assumption on uniform Δ for simple presentation. However, partial covering of a target with another agetn, the full ordering of the target does not have to hold in some cases and we use the more general version of the linear program presented in Theorem 4.7.

The proposed algorithm is run against the adversary capable of modifying the importance of each target by a fixed Δ up or down as long as the importance stays in the interval [0,99]. The explored settings of the fixed Δ range from five to ninety five. The results of the experiment are presented in Figure 3(c).

Note that unlike in the previous figures, the vertical axis shows the maximal uncovered requirement ($Curr\_DIFF$) that is defined as the optimization criterion in the problem definition in Section 3. Therefore, the lower values are better.

The results are similar to the results in the formal deception game with the suboptimal adversary. When the capabilities of the adversary are low, and it uses them in a sub-optimal heuristic way, the random exploration method of DSA being deception ignorant and maximizing the apparent importance, outperforms the proposed algorithm. However, with higher capabilities of the adversary, the proposed method is clearly superior.

7. CONCLUSION

Realistic military applications of mobile sensing agents networks, are expected to include adversaries which try to reduce the network efficiency. Previous attempts to address the problem of effectively monitoring (covering) an area using a network of mobile sensing agents did not consider such an adversary in their model.

In this paper we focused on the ability of an adversary to use means of deception (i.e. decoys, camouflage) to draw attention to less important targets and keep the more important targets uncovered. The adversary is assumed to be able to either increment or decrease the perceived importance of targets by a bounded amount.

For a single agent, we address the problem of covering one of several targets in a way that optimizes the chance to cover the most important target. To this end, we define a formal deception game and using game theoretical techniques, we derive a strategy that optimizes the position selection of the sensor in case of the worst possible real importance of the targets, consistent with the available observation. This approach results in a method which is robust against any deception and it never performs worse than a complete random selection which ignores all the (possibly misleading) information observed.

The worst scenario approach might be too careful in cases the adversary does not use its capabilities optimally. The success of the proposed method is dependent on the intensity of the deception. If the adversary uses only a very small part of its deception ability, then ignoring the possibility of deception can produce the best results. With increasing amount of ability to modify the reflected importance of targets, the advantage of the proposed method over the naïve method becomes apparent.

The formally sound local method is used as the bases of a deception aware heuristic algorithm for a full scenario in which a complete network of mobile sensing agents is operating. When compared with a local distributed algorithm (DSA) which follows a random exploration approach the proposed distributed algorithm is beneficial when the adversary has large deception abilities. For the full scenario, as for the single sensor problem, for small deception capabilities, a naïve approach of selecting positions which enable the covering of the targets which appear to be most important is enough, while with increasing deception capabilities, the success of the proposed algorithm becomes apparent.

In future work we intend to extend the optimal method for small groups of k agents and apply them in $k-opt$ algorithms for the full scenario.

8. ACKNOWLEDGMENTS

This research was partially funded by AFOSR MURI award FA9550-08-1-0356, AFOSR USAF grant number FA8655-09-1-3060 and by the Ministry of Education of the Czech Republic grants ME09053 and MSM6840770038.

9. REFERENCES

[1] B. Fristedt. The deceptive number changing game, in the absence of symmetry. *International Journal of Game Theory*, 26(2):183–191, 1997.

[2] J. Hespanha, Y. Ateskan, and H. Kizilocak. Deception in non-cooperative games with partial information. In *Proceedings of the 2nd DARPA-JFACC Symposium on Advances in Enterprise Control*, 2000.

[3] M. Jain, M. E. Taylor, M. Yokoo, and M. Tambe. Dcops meet the real world: Exploring unknown reward matrices with applications to mobile sensor networks. In *Proc. Twenty-first International Joint Conference on Artificial Intelligence (IJCAI-09)*, Pasadena, CA, USA, July 2009.

[4] K. Lee. On a deception game with three boxes. *International Journal of Game Theory*, 22(2):89–95, 1993.

[5] R. T. Maheswaran, J. P. Pearce, and M. Tambe. Distributed algorithms for dcop: A graphical-game-based approach. In *Proc. Parallel and Distributed Computing Systems PDCS)*, pages 432–439, September 2004.

[6] J. Marecki, N. Schurr, M. Tambe, and P. Scerri. Dangers in multiagent rescue using defacto. In *Workshop on Safety and Security in Multiagent Systems, AAMAS '05*, 2005.

[7] J. Michael and R. Riehle. Intelligent software decoys. In *Proc. Monterey Workshop: Eng. Automation for Software Intensive Syst. Integration*, pages 178–187. Citeseer, 2001.

[8] P. Root, J. De Mot, and E. Feron. Randomized path planning with deceptive strategies. In *American Control Conference, 2005. Proceedings of the 2005*, pages 1551–1556, 2005.

[9] Y. Shoham and K. Leyton-Brown. *Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations*. Cambridge University Press New York, NY, USA, 2008.

[10] J. Spencer. A deception game. *American Mathematical Monthly*, pages 416–417, 1973.

[11] S. Waun and U. Ozguner. A coordination strategy for cooperative sensor network deception by autonomous vehicle teams. In *43rd IEEE Conference on Decision and Control, 2004. CDC*, volume 4, 2004.

[12] W. Zhang, Z. Xing, G. Wang, and L. Wittenburg. Distributed stochastic search and distributed breakout: properties, comparishon and applications to constraints optimization problems in sensor networks. *Artificial Intelligence*, 161:1-2:55–88, January 2005.

[13] R. Zivan, R. Glinton, and K. Sycara. Distributed constraint optimization for large teams of mobile sensing agents. In *International Joint Conference on Web Intelligence and Intelligent Agent Technology*, pages 347–354, 2009.