A Game Theoretic Model for Management of Mobile Sensors

L. Ronnie M. Johansson* Ning Xiong Henrik I. Christensen
Centre for autonomous systems,†
Department of Numerical Analysis and Computer Science,
The Royal Institute of Technology (KTH),
SE-100 44 Stockholm, Sweden

Abstract – *This paper suggests a framework for multi-sensor multi-target tracking with mobile sensors. Sensors negotiate over which targets to track (possibly sharing targets to benefit from data fusion technology) using a game theory based algorithm. Sensors' preferences over negotiation offers are articulated with individual utility functions which encompass both information gain and directional derivative. An approach to consider terrain effects on mobile sensors is also explained. Simulation results show that the negotiation algorithm has interesting advantages compared to a greedy algorithm that seeks to optimise information gain without consideration to derivatives. We notice that the negotiation procedure forces sensors to share targets, while improving robustness to sensor failure. Sensors also tend to proactively reconsider their target assignments for long-term improved information gain.*

Keywords: Sensor management, mobile sensors, negotiation, game theory

1 Introduction

Mobile sensing resources (or *mobile sensors* for short) provide a flexible aid to decision support systems for decision-making in dynamic, spatially extensive environments. Their sensing capabilities contribute with observations to the decision support system and their mobility allow them to adapt to a changing world state and altered mission requirements.

Sensors is a limited resource and to achieve good performance in a system with mobile sensors, allocation and use of sensors is a key aspect to consider. *Sensor management* is the process that aims at controlling sensors to improve overall system performance [1]. Typical factors of concern for a practical sensor management system are probability of target detection, track/identification accuracy, probability of loss-of-track, probability of survival, probability of target kill, etc [2].

One aspect of managing mobile sensors is coordination of their actions. Choosing a *centralised* approach to coordinate the system promises to provide the system with optimal coordination. However, such a system is both vulnerable (e.g., if the centralised control node is destroyed, the whole system will fail) and slow (e.g., sensors have to await orders from the centralised control). Decentralised control, on the other hand, assumes that the system is mainly controlled by its components (e.g., mobile sensors), allowing it to "degrade gracefully" if some of its components fails. However, achieving good performance with decentralised control is a, by far, greater problem.

Distributed artificial intelligence (DAI) is a research field that concerns itself with coordinated interaction among distributed entities, known as *agents* [3]. *Game theory*, constituting a toolbox of methods for analysing interactions between decision makers [4], has attracted a lot of attention from the DAI community.

Game theory offers models for distributed allocation of resources and provides at the same time mechanisms to handle uncertainty. An important subtopic of game theory is *negotiation*. As part of negotiation there are ways to generate multi-objective optimisation results that are at least Pareto optimal. At the same time, these methods allow for robust handling of game/agent configurations which makes it robust to jamming and use of sensors with limited availability.

Works in DAI seldom consider uncertainties [5] such as those imposed by the physical world (e.g., estimation errors) which are inherent to target tracking applications. Noteworthy, recent exceptions concerning target tracking include [6] and [7]. In [6], static sensors form coalitions (groups) where each coalition track a certain target. The members of a coalition fuse their measurements to improve target state estimation. In [7], mobile sensors form coalitions to track targets, each sensor capable of sensing one target at a time. Movements of sensors are decided by a hierarchy of coalition leaders, each responsible for a certain geographical area.

In this paper, we address the issue of management of mobile sensors for the multi-sensor target tracking problem. Unlike [7], we do not consider the same level of comprehensive mission goals and let the sensors of our *target tracking system* handle their own motion by themselves, rather

*Tel: +46-8-790-6724, Fax: +46-8-723-0302, E-mail: rjo@nada.kth.se
†This work was financially supported by the Swedish Defence Research Agency, Division of Command and Control.

than appointing this task to some external process.

The management, in our approach, is performed using negotiation models from game theory. We utilise an algorithm for agent negotiation which we have previously developed and evaluated [8]. In the previous work, sensor agents negotiated about which targets to track, dividing the set of targets among themselves. Sensors were static, but now we apply the same algorithm to the case with mobile sensors and allow sensors to share targets. Our work constitutes a framework for future studies of management of distributed mobile sensors in the face of uncertainties and sensor failures.

Section 2 presents the primary objectives of this work, including management of mobile sensors and sharing of targets. Section 3 explains the negotiation procedure and its utility functions. Section 4 presents some results of using the negotiation strategy and, finally, Section 5 concludes and suggests future research.

2 Primary objectives

In this paper, we extend the preceding work [8] by incorporating the following three aspects:

Mobile sensors We allow sensors to move to increase sensor performance. We further allow the characteristics of the terrain to affect the *preferred direction of motion*.

Shared targets We extend the previous work by allowing sensors to track the same targets (previously, the targets were divided between the sensors). Through use of multiple sensors tracking the same target, it is possible to improve the performance on state estimates as typically found in the multi-sensor tracking and multi-sensor fusion literature (e.g., [9]). Here, this problem is studied in the context of target assignment and performance optimisation.

Performance loss when tracking many targets We model that the measurement performance on each target tracked by a sensor decreases with the number of targets tracked by the same sensor. The reason is, of course, that a sensor has limited time and resources for its measurements and if it has to track more targets and divide its resources among the targets, then also the measurement error covariance will increase for every target (hence, decreasing sensor performance on every target).

In order to allow the mobility of sensors to have any effect, we further assume that sensor platforms have the ability to move at a speed that is comparable to the speed of the targets.

3 Utility and Negotiation for Mobile sensors

Sensor agents negotiate by making offers that the other agents might accept or reject. As in the previous work, an offer, o, is a specification of allocations, that assigns groups of targets to sensors. Unlike the previous work, the target groups may overlap. From the preceding work, we bring the negotiation procedure and the notions of *reward* and *sensor information gain*. The negotiation procedure was shown to have the interesting property that the first proposing agent of a round of negotiations could, given that the other agents were benign (i.e., cooperative), propose an offer that all other agents immediately would accept (hence, minimising usage of communication resources). To evaluate offers, agents used a utility function based on the concepts of reward and sensor information gain. The information gain was

$$g_j(i) = \|H_{ij}^T R_{ij}^{-1} H_{ij}\|, \tag{1}$$

i.e., the information gain on target j by a sensor i. The concept of information gain originates from the target state estimation error covariance in Kalman filter theory (e.g., in [10]). The reward function of an agent integrated the gains on all targets into a representative value for the performance of the sensor.

For negotiation about target allocation of mobile sensors, we consider both the reward for each sensor as well as its *directional derivative* in the preferred direction of motion. The reward, as we will see, is calculated somewhat differently than in the previous work and does not immediately yield the negotiation utility. The preferred direction of a sensor platform is the spatial direction in which the sensor would like to travel. When we do not consider terrain characteristics, the preferred direction will simply coincide with the gradient of the reward function.

In the next section, we will first present an approach to consider mobility in the negotiation. In the subsequent sections 3.2-3.4, we will discuss how to calculate both the reward for the novel considerations of overlapping target groups and decreased tracking performance, and the preferred direction. We also address the resulting multi-objective optimisation problem.

3.1 Negotiation

Before we start to discuss the details about reward and directional derivative, we will for this work assume that every sensor agent has the required information and is capable of calculating both objectives for all sensors.[1] Hence, given a sensor agent $i \in S$ (S being the set of all sensor agents) and an offer of allocations of sensors to targets $o \in O$ (O being the set of offers), we can calculate reward $r_i \in \mathbb{R}$ and directional derivative $r'_{i,\delta} \in \mathbb{R}$, i.e., a sensor and an offer yield a reward and preferred direction, $S \times O \rightarrow \mathbb{R} \times \mathbb{R}$. Here $\delta \in \Delta$ is the preferred direction, and Δ the set of unit vectors.

We want to consider both factors, reward and derivative (measured as change in reward per length unit), simultaneously to acquire a combined utility metric for our negotiation. The problem we are facing is that of multi-objective

[1] In a practical application, the complete knowledge is not going to be available to all sensors, but for an initial study it is convenient to make this assumption.

optimisation. Whereas elaborate approaches to this problem has been proposed (such as [11]), in this work we prefer to study the results of an approach that does not suggest a preference of one factor over the other. (Hence, it might not work optimally for every application, but is expected to work well for every application.) We order the offers only according to *dominance*.

A sensor agent, i, will prefer an offer o_1 to another offer o_2, $o_1 \succ_i o_2$, if and only if o_1 dominates o_2. An offer o_1 can only dominate another offer o_2 if one of the reward and directional derivative values of o_1 is greater than the corresponding value of o_2 and the other one at least as great as its counterpart, i.e., $r_i(o_1) \geq r_i(o_2)$ and $r'_{i,\delta}(o_1) \geq r'_{i,\delta}(o_2)$ and at least one of the inequalities should be strict. If neither o_1 nor o_2 dominates the other, we write $o_1 \sim_i o_2$.

We elaborate further on the topic of dominance. Figure 1 shows twenty offers, here depicted with circles, plotted in a graph according to the reward and derivative in the preferred direction of a certain agent (in general, of course, the graph will look different for every agent). We find that there are, in this example, five offers that are not dominated by any other offer. We conclude that these are the "best" offers the agent could get. We call the set (or *class*) of these the *offers of the first order*. We iteratively classify the rest of the offers, knowing that an offer o_k, which is dominated by an offer o_l of order l, will be a member of order $l+1$ or greater. Each offer in Figure 1 belongs to one of five orders and the members of each order are connected to each other with dashed lines for illustration.

A more formal definition of class of offers for a particular agent is as follows.

Definition: Class of offers All pairs of offers (o_1, o_2), $o_1, o_2 \in O$, that fulfill the condition that $o_1 \sim o_2 \wedge \neg \exists o_m \in O\,[(o_m \sim o_j \wedge o_m \succ o_k)]$ for $j \neq k$ and $j,k \in \{1,2\}$ are said to belong to the same class of offers.

A class may not be empty, but may contain a single offer o_s iff $\forall o_j \exists o_k\,[o_j \sim o_s \wedge o_j \sim o_k \rightarrow o_k \succ o_s \vee o_k \prec o_s]$. In order to strictly define class order, we first define the notion of *class dominance*.

Definition: Class dominance A class of offers C_a is said to dominate another class C_b, $C_a \succ C_b$, iff $\exists o_a \exists o_b\,[o_a \succ o_b]$, $o_a \in C_a, o_b \in C_b$.

We use the following recursive definition to define order of class.

Definition: Class of first order A class C of offers is said to be of the first order iff none of its offers are dominated by another offer, $\neg \exists o_m \in O \setminus C\,[o_m \succ o_j]$ for all $o_j \in C$.

Definition: Class of kth order A class of offers C is said to be of order k iff its members are dominated only by members of classes of order k and less.

Now, we will express the utility function for sensors. Using our notion of orders we can assign a utility value to the offers for all agents. Furthermore, according to the negotiation procedure described in [8], the utility of an offer accepted at time $t+1$ is always less valuable than the same offer accepted at time t. Therefore we need to construct a utility function that is dependent on the time step of the negotiation:

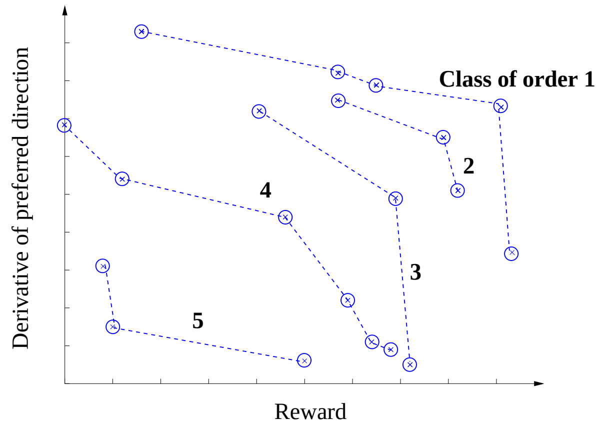


Figure 1: Twenty offers plotted according to derivative of preferred direction and reward. Offers which belong to the same class are connected with lines.

$$U_i(o,t) = \alpha_U(K - t) - order_i(o), \quad \alpha_U > 0, \tag{2}$$

where $order_i(o)$ is a function that maps offers to its order for sensor agent i. Thus, offers of low order (which are desired by the agent) yield a high utility value.

The interpretation of the utility function in Equation 2 is that the agents will accept less offers the longer the negotiation continues.

The negotiation procedure in the preceding work is virtually unaffected by the extensions we make in this work. The reason for this is that all the novelties (in Section 2) have been encapsulated in the calculations of the utility function. However, the result of the negotiation will of course be quite different.

An agent that has several "best" offers to choose from should select one according to some second criterion. This could for instance comprise minimising the sum of orders for all agents, i.e., if the set of best offers is O', then the offer to select should be the one o^* that satisfies $o^* = \arg\min_{o \in O'} \sum_i order_i(o)$. Another suitable criterion could be to select the offer in O' that minimises maximum order for any sensor, i.e., $o^* = \arg\min_{o \in O'} \max_i order_i(o)$. If there are still more than one offer that fulfills the criterion, then one offer could be selected randomly.

3.2 Workload effect on tracking performance

A sensor is expected to make less certain measurements for each target if it tracks many targets than if it tracks only a few. Let us assume that sensors have some sort of resource (e.g., time, energy, money, samplings) that they can utilise to make measurements. The maximum amount of this resource available to a sensor i for a time unit is $\rho_{i,max}$ and the amount it chooses to use to track some target j is denoted ρ_{ij}. We model that the measurement noise v_{ij} in the Kalman measurement model is dependent on the dedicated resource amount ρ_{ij}. The measurement noise is Gaussian, i.e., $v_{ij} \sim N(\mathbf{0}, R)$, with zero mean and the measurement error covariance matrix R.

R is composed of standard deviation functions for the measurement variables. The functions, $\sigma_l(\rho)$, will take a minimum, $\sigma_{min,l} \geq 0$, for $\rho = \rho_{max}$ and will increase towards infinity when the dedicated resource decreases towards ρ_{min}, $\lim_{\rho \to \rho_{min}(j)} \sigma_l(\rho) \to \infty$, where $\rho_{min}(j)$ is the minimum resource amount necessary to track target j.

Hence, a varying workload on a sensor will affect the standard deviation and the measurement error covariance matrix, which in turn will have effect on the refined sensor gain expression which we will discuss in the next section.

Note that using this model, we allow sensors to allocate different amounts of resources to different targets.

3.3 Target allocation

In the preceding work, the specific task was studied where every target was tracked by exactly one sensor. In this work, we relax that restriction and allow sensors to "share" targets. Thus, we are able to reduce uncertainty by fusing measurements from different sensors and get a higher grade of sensor usage than in the case of disjoint target groups.

Our approach here to determine the reward for every sensor, r_i, is to divide the *total reward* on every target, $\sum_j r_j(S_j)$ (S_j being the set of sensors tracking target j), among the sensors in S_j proportionally to their individual contribution.

We define the reward on every target to be

$$r_j(S_j) \triangleq 1 - e^{-\alpha_j g_j(S_j)}, \quad \alpha_j > 0 \ , \tag{3}$$

where

$$g_j(S_j) = \sum_{k \in S_j} \|H_{kj}^T R_{kj}^{-1} H_{kj}\|, \tag{4}$$

i.e., the total information gain on target j. Here, we might want to replace g_j with some measure from information theory. Previously, we used sensor information gain, but now, as we allow multi-sensor fusion, we define $g_j(S_j)$ as above and notice that whenever S_j contains a single sensor i $g_j(S_j) = g_j(i)$ (equivalent to Equation 1).

Now, the *net reward* for every sensor (similarly expressed as in the previous work) is

$$r_i^{net}(D_i) \triangleq \alpha_i + (1 - \alpha_i) r_i^m(D_i), \quad 0 \leq \alpha_i \leq 1 \ , \tag{5}$$

where D_i is the group of targets tracked by sensor i, α_i reflects the willingness of a sensor agent to compromise about offers, and the measurement reward is

$$r_i^m(D_i) \triangleq \sum_{j \in D_i} \gamma_{ij} r_j(S_j) \tag{6}$$

and

$$\gamma_{ij} \triangleq \frac{g_j(i)}{g_j(S_j)} = \frac{\|H_{ij}^T R_{ij}^{-1} H_{ij}\|}{\sum_{i \in S_j} \|H_i^T R_{ij}^{-1} H_i\|}, \tag{7}$$

i.e., the relative contribution of sensor i to the state estimate of target j.

This definition of sensor reward, $r_i^m(D_i)$, has the effect that the same amount of information gain from a sensor on a target will yield different rewards depending what other sensors track the same target. This makes sense since the target reward does not improve linearly with the information gain (e.g., in a target tracking application, tracking airborne targets at high speeds, to go from metre to centimetre precision in position estimates should not yield much extra reward since the improved precision can not be efficiently utilised).

3.4 Preferred direction

Given the measurement reward function, $r_i^m(D_i)$, for each sensor, the gradient can be calculated in this way:

$$grad\ r_i^m \equiv \nabla r_i^m \equiv \left(\frac{\partial r_i^m}{\partial x_i}, \frac{\partial r_i^m}{\partial y_i}\right), \tag{8}$$

where x_i and y_i are the spatial coordinates of sensor i's position.

The gradient vector points in the direction in which the reward for sensor i will increase the most.[2] This model makes the subtle (and incorrect) assumption that the targets are static. However, it is a fairly good approximation that should be refined in the future; possibly by predicting and exploiting future target states. The gradient would be the preferred direction to move for the sensor if terrain properties were not considered.[3] However, the terrain may make motion in the direction of the gradient difficult or perhaps even impossible, and a more passable path, although less rewarding, might be a better preferred direction.

Now assume we can construct a (possibly rough) terrain dependent function, which discounts the reward change in various directions. Let the *terrain function* be $t(\mathbf{p}, \mathbf{e}_\theta)$, where $\mathbf{p}$ is a two dimensional position in the environment and $\mathbf{e}_\theta$ is a unit vector, $\theta \in [0, 2\pi)$. Furthermore, let the terrain function assume values between 0 and 1, $t \in [0, 1]$. The terrain function $t(\mathbf{p}, \mathbf{e}_\theta)$ takes high values in directions where the sensor platform can easily move (such as in the direction of a good road) and low values in directions where it cannot move very well (zero in the direction of an unpassable obstacle). We assume that the value reflects the passability in the chosen direction in the following time step.

The directional derivative $r'_{\mathbf{e}_\theta}$ in any direction, $\mathbf{e}_\theta$, is simply a projection of the gradient onto $\mathbf{e}_\theta$, i.e., $r'_{\mathbf{e}_\theta} = \mathbf{e}_\theta \bullet \nabla r^m$. The parameter θ is the angle between the gradient and $\mathbf{e}_\theta$.

Now, we propose that the preferred direction, δ^*, is the unit vector that corresponds to the largest directional derivative discounted by $t(\mathbf{p}, \mathbf{e}_\theta)$, i.e.,

$$\delta^* = \arg\max_{\mathbf{e}_\theta} \left\{ t(\mathbf{p}, \mathbf{e}_\theta) \cdot r'_{\mathbf{e}_\theta} \right\}. \tag{9}$$

[2] Note that we are, in this work, only considering the current target states when calculating the gradient. Prediction of future target states to further improve the performance of the mobile sensors is left for future work.

[3] Hence, in the case of airborne sensors, the gradient would suffice as a preferred direction.

We now expand our field of view to study the preferred directions from a macro perspective. Figure 2 shows the directional derivatives in various positions in the plane. A target in position $(400,350)$ (the small "x") attracts a sensor platform. An obstacle (representing almost unpassable terrain) has been positioned to the left in figure. We note that the derivatives are small in the periphery and close to the target, and large in between (these are characteristics of the measurement reward function in Equation 6). The preferred directions direct the sensor platform away from the obstacle, while trying to preserve a course towards the target. For instance, along the upper and lower edges of the obstacle, the preferred directions are along the edge of the obstacle rather than into the obstacle.

Even though the approach with terrain functions presented here looks nice in this example, it is indeed short-sighted. There is a risk that sensor platforms get stuck behind obstacles. However, this does not necessarily mean that the tracking will fail, rather it means that the current allocation has been given a new value which will possibly affect the outcome in the next round of negotiations (i.e., another allocation, with a better preferred direction, might be a more appealing alternative).

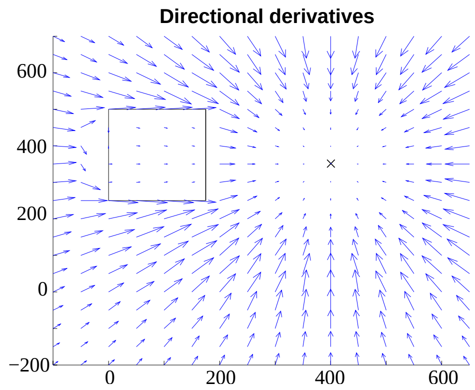


Figure 2: An obstacle (representing very rough terrain) is situated in the left part of the figure. The generated preferred directions tries to steer the sensor platform away from the obstacle while preserving a course towards the target (the 'x').

4 Experimental results

For our simulations,[4] we assume that the standard deviation, σ_{tot}, of the measurement noise covariance, R, is equal for every measurement component and tracked target. We, furthermore, assume that it increases inversely linearly with the dedicated relative resource amount, i.e., the standard deviation is scaled by a factor $\left(\frac{\rho}{\rho_{max}}\right)^{-1}$, and quadratically with the Euclidean distance d between target and sensor. If the tracking resource, discussed in Section 3.2, is divided evenly between n tracked targets, the resource amount used to track each of the targets is $\rho(n) = \rho_{max}/n$, yielding the scale factor $\left(\frac{\rho_{max}/n}{\rho_{max}}\right)^{-1} = n$ for the standard deviation. From this discussion, we suggest the following standard deviation expression for our experiments

$$\sigma_{tot} = \sigma_{min} \cdot n \cdot (1 + cd^2). \tag{10}$$

The first two factors are always greater than zero and $d \geq 0$. The coefficient $c > 0$ controls how greatly the distance from sensor to target affects the measurement error covariance.

We require that the target tracking system always tracks all targets (i.e., negotiation offers that do not include assignments to all targets will be ignored by all sensor agents).

4.1 Target tracking with mobile sensors

For the simulations with mobile sensors, we use the utility function in Equation 2, but we will for now assume that the terrain has no effect on the negotiation.

For evaluation, we devise a comparison algorithm, a "greedy" tracker (`G-tracker`), which operates independently of the negotiation-based tracker (`N-tracker`). We let the `G-tracker` reconsider the sensor to target assignment as often as the `N-tracker` does. After having selected the most optimal assignment, the sensors travel, at full speed, in the direction of the gradient. Whereas that seems reasonable, we will see examples where the `G-tracker` encounters difficulties.

In our first simulation, we want to know whether our reward function makes sensors try to fixate one target or if they tend to locate themselves where measurement performance on all targets is good. In Figure 3, two sensors track four targets. In this and the following figures that depict snapshots of target tracking, the crosses are targets, the tiny circles are the mobile sensors, and the line that extends from the centre of each sensor indicates the current direction of motion of the sensor (it does not, however, indicate the speed of the sensor). Additionally, in some of the figures, dotted lines are drawn from sensors to targets. These lines clarify which sensors are tracking which targets.

The simulation starts at time $t = t_1$, and at this time the targets are divided between the two sensors in such a way that the upper sensor is willing to track the two upper targets and the lower sensor is willing to track the two lower targets. The upper targets are moving upwards and the lower targets are moving downwards. We see that the sensors, which in this simulation have the ability to catch up with the targets, prefer to situate themselves in between the targets.

In our next experiment, we study a scenario where the `G-tracker` runs into problems. In this case, sensor s_1 (in Figure 4(a)) wants to track the targets τ_1 and τ_2. However, they move in opposite directions, leaving s_1 with a resulting zero gradient, i.e., s_1 gets stuck while the targets move away (as seen in Figure 4(b)). Sensor s_2 on the right has a similar problem since its targets are also moving in opposite directions. After a while, however, the `G-tracker` assigns targets τ_1 and τ_3 to sensor s_1 and the others to s_2, allowing sensor s_1 to escape from its deadlock. If we align targets

[4] In this paper, merely snapshots of simulations are shown. However, full animations are available at this URL: `http://www.nada.kth.se/~rjo/pubs/mobile/anim/`.

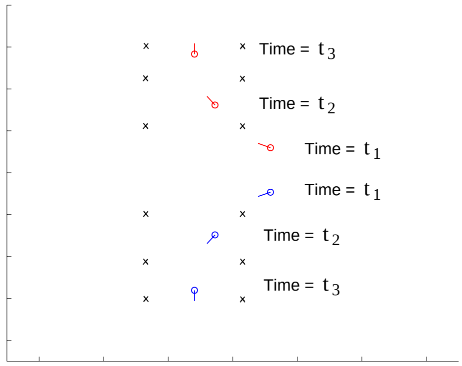
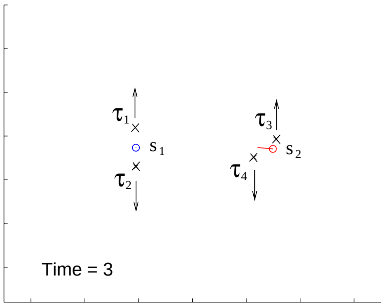
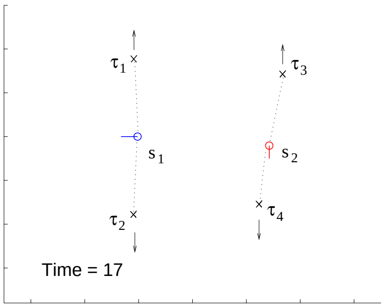


Figure 3: The Figure shows three superimposed snapshots, at times t_1, t_2 and t_3 ($t_1 < t_2 < t_3$), of a scenario where two sensors track two targets each.



(a)



(b)

Figure 4: (a) In this scenario, two sensors s_1 and s_2 track four targets τ_1 to τ_4. Targets τ_1 and τ_3 are moving upwards and τ_2 and τ_4 downwards. Initially, the `G-tracker` assigns τ_1 and τ_2 to s_1 and τ_3 and τ_4 to s_2. (b) After some time, the targets have moved, but due to the greedy allocation of targets to sensors, the sensors are stuck between their assigned targets and have hardly moved.

τ_3 and τ_4 with sensor s_2 and rerun the simulation, we can actually make both sensors get stuck forever.

The `N-tracker`, run on the same scenario, yields a more appealing result. To begin with, we see that the negotiation brings about a somewhat surprising assignment of targets to sensors (Figure 5(a)); s_1 tracks τ_3 and τ_4, and s_2 the other two, contrary to the allocation of the `G-tracker` (see once again Figure 4(a)). The reason is of course that the "greedy" allocation yields very low directional derivatives which allows the `N-tracker` to reach other solutions.

After a short while, sensor s_1 starts to follow the targets τ_2 and τ_4 that are moving downwards, and the other two are followed by sensor s_2 (Figure 5(b)).

In Figure 6, we compare the results of the `N-tracker` and `G-tracker` in terms of reward. At time $t = 10$, the `N-tracker` decides that sensor s_1 should track targets τ_2 and τ_4 and quickly receives a total reward which is greater than that of the `G-tracker`. At time $t = 27$, also the `G-tracker` decides that one sensor should track the targets moving upwards and the other the ones going downwards. That explains the negative slope of the curve in the end. However, as we can see from the rewards in the figure, the `G-tracker` is unable to catch up with the `N-tracker`. Since the targets in this scenario are allowed to travel at a higher speed than the sensors, the reward drops rapidly and at time $t = 40$ and beyond, both algorithms receive very low rewards.

In the final experiment of this section, we study the scenario in Figure 7. Three sensors track four targets. The targets travel in two pairs, one pair entering from the left and one from the right. The flight paths of all targets are shown with dashed lines in the figure.

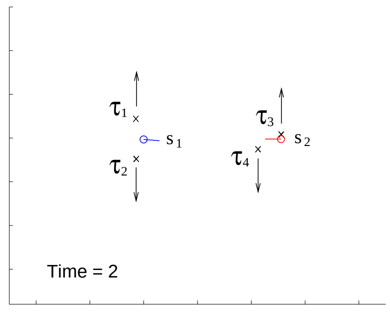
We run both the `G-tracker` and the `N-tracker` and compare the results in Table 1. We compare three values for the two trackers. *Reward* is simply the average reward of every round of negotiations, *redundancy* is the average number of targets being tracked by one or more sensors, and *lost targets* is the average number of targets without any sensor tracking them if one of the sensors fails. We see that the `N-tracker` is defeated in measurement accuracy (its average reward was 90% of that of the `G-tracker`). However, the `N-tracker` instead impresses by its robustness with an average of 1.39 targets being tracked by one or more sensors and an average of 0.87 (27% better than the result of the `G-tracker`) of lost targets if one sensor is lost. The reason for this result is that the sensors, through the negotiation, are forced to share targets with each other, and, hence, yield better robustness for the target tracking system as a whole.

Table 1: Comparison between `G-tracker` and `N-tracker`

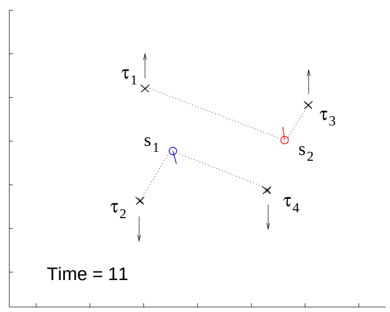
	G-tracker	**N-tracker**	**Relative**
Reward	3.7372	3.3765	0.90
Redundancy	0.4510	1.3922	3.09
Lost targets	1.1830	0.8693	0.73

4.2 Mobile tracking with terrain considerations

Until now, we have not considered terrain effects on mobile sensors in our experiments. Since it is highly unlikely



(a)



(b)

Figure 5: (a) Initially, the negotiation algorithm assigns targets τ_3 and τ_4 to sensor s_1 and the rest to sensor s_2. (b) After some time, the negotiation algorithm assigns targets τ_2 and τ_4 to sensor s_1 and the rest to sensor s_2.

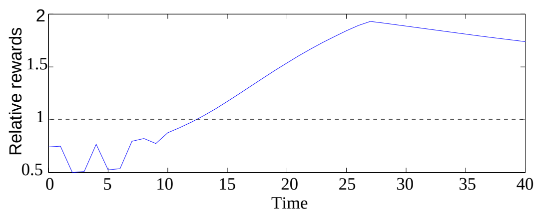


Figure 6: This graph compares the total rewards of the `G-tracker` and the `N-tracker` to each other. The relative reward of the `N-tracker` compared to the `G-tracker` has been plotted.

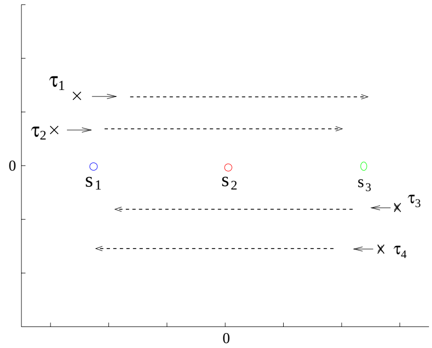


Figure 7: In this scenario, three mobile sensors (s_1, s_2 and s_3) track four targets (τ_1 to τ_4).

that the designer of a mobile sensor system can expect a homogeneous environment, we need to consider varying terrain and its effects. In Section 3.4, we discussed how a so-called terrain function can be used to discount the directional derivative generated by a certain assignment.

In the scenario in Figure 8, we have put an *obstacle* into the environment. This obstacle has the property that when a mobile sensor tries to cross it, the maximum speed of the sensor reduces drastically. Such an obstacle represents, for instance, rough terrain or a steep hill. In this example, the speed reduces to 30% of the maximum speed it could achieve in an ideal terrain. Close to the obstacle, the terrain function discounts directional derivatives that lead into the obstacle.

We notice that the `G-tracker`, which does not consider terrain, leads the sensors straight into the obstacle, as shown in Figure 9(a). As a result of this, the sensors lose touch with the targets. In the case of the negotiation algorithm, the sensors switch targets close to the border of the obstacle, as shown in Figure 9(b). One reason for this is that offers that give directions that lead into the obstacle get small derivatives and are suppressed.

5 Conclusion and future work

In this paper, we have presented a game theoretic model for assigning targets to mobile sensors. Sensor agents negotiate by proposing offers of allocations that involve all sensors. Each agent can evaluate each offer to decide its individual utility.

We showed, in the experiments in Section 4, two interesting properties of our negotiation algorithm: first, the negotiation forces sensors to share targets, improving robustness to the target tracking system (e.g., the scenario in Figure 7). Secondly, considering directional derivatives allow sensors to proactively reconsider target assignments, possibly improving long-term information gain (e.g., as in Figures 5(b) and 9(b)).

Further studies should investigate under what circumstances these properties imply advantages to the target tracking system. With the support of these early results, we

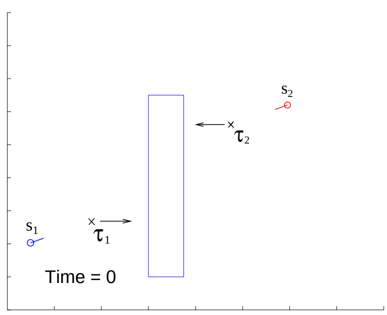
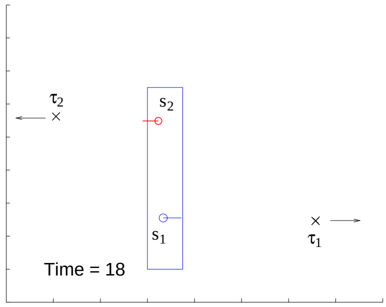
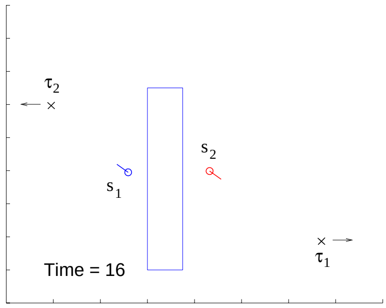


Figure 8: Initially, sensor s_1 tracks target τ_1 and sensor s_2 target τ_2. The rectangle represents an area which slows down mobile sensors that enter it.



(a)



(b)

Figure 9: (a) The `G-tracker` does not consider terrain and leads the sensors into the obstacle, where they are slowed down considerably. (b) The negotiation algorithm decides to switch targets between the sensors instead.

anticipate interesting discoveries in our future exploration of negotiation-based, distributed sensor management.

Some of the most salient, concrete directions for future studies are:

- introduction of uncertainty (e.g., in target or sensor state) into the negotiations,
- prediction of (near) future target and sensor states to improve tracking performance,
- to explore and devise a policy to select negotiation strategy depending on the state of the environment.

References

[1] G. W. Ng and K. H. Ng, *Sensor management - what, why and how*, Information Fusion, Vol 1, pp. 67–75, 2000.

[2] Ning Xiong and Per Svensson, *Sensor management for information fusion - issues and approaches*, Information Fusion, Vol 3, pp. 163–186, 2002.

[3] Gerhard I. Weiss, editor, *Multiagent Systems - A Modern Approach to Distributed Artificial Intelligence*. MIT Press, 1999.

[4] Martin J. Osborne and Ariel Rubinstein, *A Course in Game Theory*. MIT Press, 1994.

[5] Y. Uny Cao, Alex S. Fukunaga, and Andrew B. Kahng, *Cooperative mobile robotics: Antecedents and directions*, Autonomous Robots, Vol 4, pp. 7–27, 1997.

[6] Pierre Dodin and Vincent Nimier, *Distributed resource allocation under communication constraints*, In Proceedings, 4th International Conference on Information Fusion, International Society of Information Fusion, 2001.

[7] Jr. Charles L. Ortiz, Eric Hsu, Marie desJardins, Timothy Rauenbusch, Barbara Grosz, Osher Yadgar, and Sarit Kraus, *Incremental negotiation and coalition formation for resource-bounded agents*, AAAI Fall Symposium on Negotiation Methods for Autonomous Cooperative Systems, Preliminary report, 2001.

[8] Ning Xiong, Henrik I. Christensen, and Per Svensson, *Negotiation of target distribution enhancing system survivability*, Report CVAP-274, Royal Institute of Technology, Numerical Analysis and Computer Science, Computer Vision and Active Perception Laboratory, 100 44 Stockholm, SWEDEN, 2003. http://www.nada.kth.se/cvap/cvaplop/lop-cvap.html, submitted to IEEE Transactions on Systems, Man, and Cybernetics.

[9] Yaakov Bar-Shalom and Thomas E. Fortmann, *Tracking and Data Association*. Academic Press, 1988.

[10] Michael Kalandros and Lucy Y. Pao, *Controlling target estimate covariance in centralized multisensor systems*, In Proceedings of American Control Conference, pp. 2749–2753, 1998.

[11] Carlos M. Fonseca and Peter J. Fleming, *Multiobjective optimization and multiple constraint handling with evolutionary algorithms-part I: A unified formulation*, IEEE Transactions on Systems, Man, and Cybernetics, Vol 28, No. 1, pp. 473–509, January 1998.