Coordinated Space Domain Awareness as an Optimized Commodity Market

Przemek Wozniak
Svetlana Riabova, Hassan Hijazi
Los Alamos National Laboratory

ABSTRACT

The convergence of autonomous robotic hardware and distributed sensor ecosystems is revolutionizing surveillance technology and observational science. Next generation systems for Space Domain Awareness will deploy networks of intelligent agents acting in the environment, requiring a mix of collaborative and individual behavior to accomplish their mission. While numerous approaches are being explored, this long-term vision presents often unique challenges. Autonomous Tip & Cue systems are often the only way to ”catch” and characterize emerging events of interest before it is too late to respond or there are too many events. The key missing piece is efficient coordination that can quantify the utility of information and optimize the overall outcome. Despite rapid progress on multiple fronts, this problem remains one of the main barriers to a wider adoption of such systems.

Here we present a solution based on the idea of using a market-place model to represent interactions within distributed multi-agent sensor networks designed to coordinate sensor ecosystems with diverse sensing capabilities, but limited resources, and no strict centralized control. We then apply this approach to the problem of Space Traffic Management with globally distributed sensors tracking a large population of space objects and minimizing the overall uncertainty and collision risk, while simultaneously optimizing individual objectives of all agents. We also demonstrate the gain in total utility from coordinated data collections compared to uncoordinated behavior. Typically, agents have limited information toward satisfying their objectives, which creates a strong incentive to trade with other agents. However, at any given time there is no obligation to participate in the market, allowing agents to retain autonomy and pursue their own goals.

The system consists of agents that own private sensors and a coordination authority (CA), that operates the exchange market for trading observations, but does not own any sensors. Sensor trade creates possibilities for the agents to observe some objects of interest and engage additional capabilities, which are not accessible from their own sensors. The CA, however, does not have any research agenda and is acting as a market regulator by setting the exchange prices to reduce the total uncertainty associated with the population of space objects. The presence of CA allows agents to avoid sharing private information (observation utilities) with each other. While the market approach is introduced to enable effective collaboration between autonomous agents, it can also be used in systems with centralized control as a way of managing complexity.

The proposed market system features a two-level hierarchy, which can be viewed as a bilevel game where the CA acts as a leader by setting the price for exchanged observations and the agents (followers) react to that pricing decision by committing to certain exchanges based purely on their interests. In this kind of game, each player solves their own optimization problem, the leader tries to generate the best outcome while accounting for the followers’ response, and the followers act in the space parameterized by the leader maximizing their individual benefit. The knowledge of agents’ utilities allows the CA to reconstruct their individual optimization problems and solve a bilevel problem predicting the response to the pricing policy.

A key aspect of bilevel optimization is that given a leader’s decision, the followers always react optimally, making these models a powerful tool to capture the relationship between agents in a market. Certain caveats are also discussed. The model features a mechanism to ensure fair observation pricing and agents are guaranteed that the resulting sensor assignment benefits their research programs. The proposed approach requires solving a mixed-integer optimization problem, which provides a globally optimal solution and is known to be NP-hard. While the optimal solution can in principle be found, in practice it is often preferred to obtain a near-optimal solution in a reasonable time.

1. INTRODUCTION

The convergence of autonomous robotic hardware enabled by Artificial Intelligence (AI) and distributed sensor ecosystems fostered by the Internet of Things (IoT) is revolutionizing surveillance technology and observational science [4, 19]. Next generation systems for Space Domain Awareness will deploy networks of intelligent agents acting in the environment, requiring a mix of collaborative and individual behavior to accomplish their mission. While numerous approaches are being explored, this long-term vision presents often unique challenges. Autonomous Tip & Cue systems are often the only way to "catch" and characterize emerging events of interest before it is too late to respond or there are too many events. Many elements of this approach are ready for experimental work or deployment: event recognition, required edge computing, communication protocols, and executing simple responses. The key missing piece is efficient coordination that can quantify the utility of information and optimize the overall outcome. Despite rapid progress on multiple fronts, effective coordination between agents remains one of the main barriers to wider adoption. This is especially difficult in heterogeneous networks of sensors with diverse capabilities and operational modes that must be matched and dynamically allocated as conditions evolve.

In this paper we present a solution based on the idea of using a market-place model to represent interactions within distributed multi-agent sensor networks [1] designed to coordinate activity of sensor ecosystems with diverse sensing capabilities, but limited resources, and no strict centralized control. We then apply this approach to the problem of Space Traffic Management with globally distributed sensor networks tracking a large population of space objects and minimizing the overall level of uncertainty and collision risk, while simultaneously optimizing individual objectives of all agents. In our collaborative sensor network architecture, the overarching assumption is that most of the time, none of the agents can obtain enough information on their own to satisfy its objectives. This assumption is strongly supported by the experience with actual sensors deployed in the real world. Therefore, there is a strong incentive to participate in the market by trading some information with other agents, but there is no obligation. There is an inherent uncertainty in how the system operates and negotiates the next actions to be undertaken.

Our approach builds on the seminal work in Contract Nets [2, 22, 23], which mimic the interaction between a general contractor using subcontractors to execute tasks in exchange for a reward. However, the proposed solution goes beyond the limits of strict command and control, allowing agents to retain the autonomy to make their own decisions and pursue their own goals. The resulting resource allocation shares similarities with auctions [18] that promote collaborative agent behavior by a carefully chosen negotiation mechanism [3].

2. RELATED WORK AND EXISTING CHALLENGES

Sensor tasking is a process of deciding which observation target a sensor needs to observe at a particular slot of time. It has been one of the key problems in SDA since the beginning of space object catalogues. Usually, there is a trade-off between the number of objects observed, observation quality and the probability of detecting new objects. A variety of models with different degrees of complexity have been developed to perform sensor tasking, starting with heuristics rules and branching into a fusion of information theory, optimization and deep learning approaches. Sensors can generally be used for two types of activity: surveys and follow-up observations. Survey observations are aimed at discovering new objects, while follow-up observations are made to reduce the uncertainty associated with already-known objects.

An early work on recovering trajectories with follow-up observations scheduled with simulation was presented by Musci et al. [16]. Hill et al. [8] used covariance as a metric for scheduling observations in time when tasking and scheduling are performed separately. Williams et al. [26] proposed coupling sensor assignment with the observation process in a closed-loop algorithm using Fischer information gain as a tasking metric and Lyapunov exponents to prevent divergence in uncertainty for some objects. Jaunzemis et al. [11] applied the Dempster-Schafer theory to develop a hypothesis-testing framework based on the evidence from the data which was later used as a basis for sensor tasking optimization. Heuristic rules were formalised in an optimization problem by Frueh et al. [5] and a computationally fast near-optimal solution was also presented. Nastasi and Black [17] modelled dynamic sensor tasking as a partially observed Markov decision process using Fischer information gain and largest Lyapunov exponent as tasking metrics.

Meta-heuristic approaches were presented in [9] and [14]. In [14] a comparison between several methods was presented and the ant colony algorithm showed the best results. Wu et al. [27] present an ensemble method utilizing both meta-heuristic and exact approaches to solve a tasking problem of observing objects on Earth from satellites located on different orbits in space. A series of deep learning approaches were developed in [10, 12, 13, 20, 21]. Linares and Furfaro [12] model the sensor management problem as a discrete Markov process and developed a reinforcement learning approach to generate a tasking sequence with two neural networks: one for generating action responses and one assessing the value of an action; a simulation setup is used for training. In [13] they extend that work with asynchronous actor-critic learning. Jang et al. [10] presented a comparison between several reinforcement learning agents trained for sensor tasking showing that a combination of models trained in different environments is resilient to unexpected scenarios. Siew et al. focused on training a single ground-based sensor to perform follow-up observations.

The majority of work in the field of sensor tasking is focused on the setups with either one sensor or a set of sensors belonging to one entity. As the orbit is getting more crowded and the research missions more ambitious, it becomes harder and harder to maintain the uncertainties associated with object positions and trajectories reasonably low without a collaborative effort. For the sake of concise presentation, this work will focus on follow-up monitoring observations only, although it is possible to extend it to include survey observations as well.

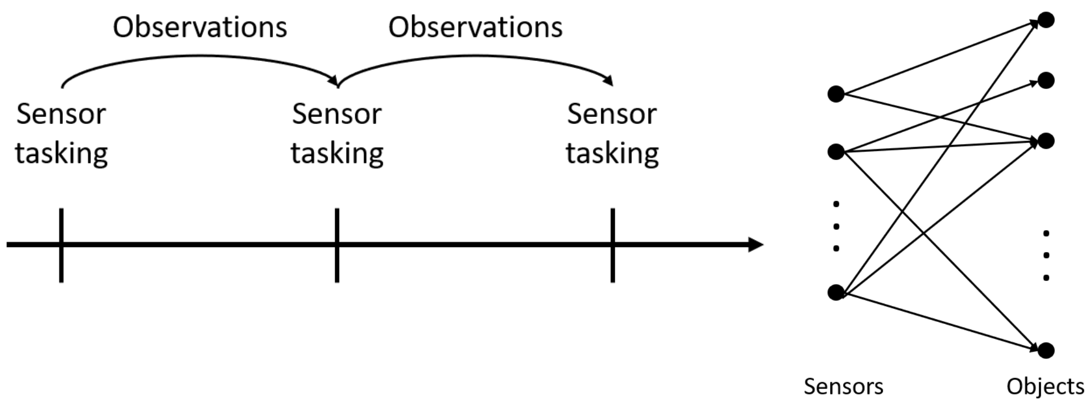
3. PROBLEM STATEMENT

Here we provide a detailed explanation of our problem setting. In this study, we consider a network that consists of agents that own and operate private sensors, a set of registered space objects and a Coordination Authority (CA) who operates the observations exchange market. The goal of this work is to create a tool that allows the agents and the CA to use all sensors in the system in a collaborative way while still respecting each agent's privacy and rights to utilize the sensors they own in a way that is most beneficial for them. We assume that the private agents are "selfish" and prioritize their own research agenda over the global goal of keeping the uncertainty of object trajectories as low as possible. The research agenda of the CA is comprised of objects that have a significant level of uncertainty associated with their trajectory due to a lack of recent observations or other reasons. It is important to note that for any agent the participation in this market exchange system is voluntary and is primarily motivated by the potential benefits for their own mission. The main attraction from an agent perspective is the opportunity to use other sensors controlled by other agents to make observations not available with their own set of sensors.

Our procedure is defined with discrete time steps. At each step, we solve an optimization problem to determine the most beneficial observations, i.e. the assignment of sensors to objects, which is followed by an information update after the observations have been made. This process is illustrated in Figure 1a. The network configuration at each step is given by a bipartite visibility graph, where the two sets of nodes represent sensors and objects, and the arcs show the potentially feasible observations (Figure 1b). In this paper we use the term *observation* to describe an arc connecting a sensor to an object. When a potential observation is carried out, the corresponding data collection results in improved object localization, which is assimilated into the calculation of sensor utility and reduction in collision risk.

The described system features K agents, a total of N sensors (distributed in some way between the agents), and M space objects to be tracked. At every time step, each agent has to decide for each of their sensors, whether they want to use it to make an observation for themselves or sell it to another agent (make an observation useful to another agent). We assume that the agents have numeric values for interest weights associated with each observation and that these values are expressed in a monetary equivalent. These weights express how much an agent values the data from a particular observation. In the market model, the weights represent how much an agent is willing to pay for an observation if they are buying or the minimum price an agent needs to be paid if they are selling.

We assume that trades between agents are carried out in the following way. The CA collects the interest information (weights) from the agents, performs market optimization, and then announces the prices to agents. Prices are set for every sensor, and by paying the price an agent obtains the right to choose which object a given sensor should observe. The owners then make observations and provide the buyers and CA with the data. As the goal of the CA is to reduce the uncertainty associated with objects, we assume that the agents agree to share all their observations with the CA (but not with each other) to contribute to this global effort in SDA. Thus, the CA only need to buy observations when no agent has committed to those beneficial data collects. We will use the term *buy a sensor* or *sell a sensor* to indicate that the buyer gets the choose which object the sensor is used for in the current optimization period (time step).



(a) Iterative process

(b) Visibility graph

Fig. 1: Rolling horizon optimization. The assignment of sensors to objects is optimized at every time step, given all information available at that time (left). The visibility graph specifies all possible assignments (potential observations) and is also computed for every time step (right).

3.1 An illustrative example

An example of a feasible assignment is presented in Figure 2. In this example, we have the CA, three agents ($K = 3$), six sensors ($N = 6$), and seven objects ($M = 7$). The first agent (blue) has two sensors, the second agent (orange) has one sensor and the third agent (green) has three sensors. The CA does not have any sensors but can still buy observations from other agents. In the presented assignment scheme, Agent 1 sells an observation from their first sensor to Agent 2, which is represented by the orange arrow connecting that sensor to Object 1 which Agent 2 decided to observe. They use their second sensor to perform their own observation of Object 3. Agent 2 sells an observation from their only sensor to Agent 1 which is represented by the blue assignment arrow. Agent 3 uses one of their sensors for their own observations and sells the other two to perform an observation for the CA and Agent 2 respectively. The arrows between the agents represent the flow of money (utility) as observations are purchased and paid for.

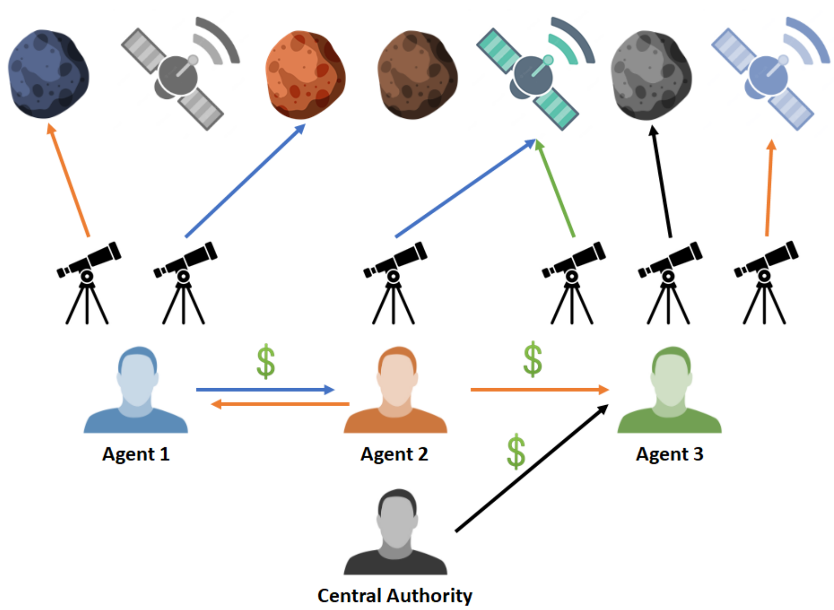


Fig. 2: An example of a feasible sensor assignment in the market model

3.2 Market modelling

In our market model, the agents retain the autonomy to manage their own sensors and the CA acts as a facilitator and intermediary between them. This way, the agents do not have to share their information with others as it is only accessible to the CA. The market interaction can be modelled as a two-level decision process, where the CA sets the prices for possible observations available for exchange, after which the agents make assignments for their sensors based on market conditions, visibility, and their own agenda. We use bilevel optimization to model this hierarchical decision process.

Bilevel optimization problems are sometimes referred to as bilevel games or Stackelberg games, as the concept originated in the work of Stackelberg [24] on leader-follower games in a market economy. Bilevel problems consist of two individual optimization problems: upper level - the leader's problem and lower level - the follower's problem. In that kind of hierarchical decision process, the leader acts first, setting parameters for the follower's problem. The follower, in their turn, solves the lower-level problem partially defined by the leader's strategy. Bilevel problems are typically solved from the leader's perspective. The leader tries to predict the follower's reaction and choose a strategy that gives them the best outcome with respect to this reaction.

A schematic representation of a bilevel game is given in Figure 3. Each combination of upper and lower-level strategies corresponds to an outcome of the game which for our application is a pricing + assignment decision as was presented in Figure 2. In our models, the leader aims to maximize the total utility received by all players in the system and also has a part of the objective associated with the objects no one else is interested in. In the context of Space Traffic Management, for example, the latter objective serves to lower the overall collision risk for a population of space objects. The followers act selfishly, maximizing their own utilities based on prices and other information available to them.

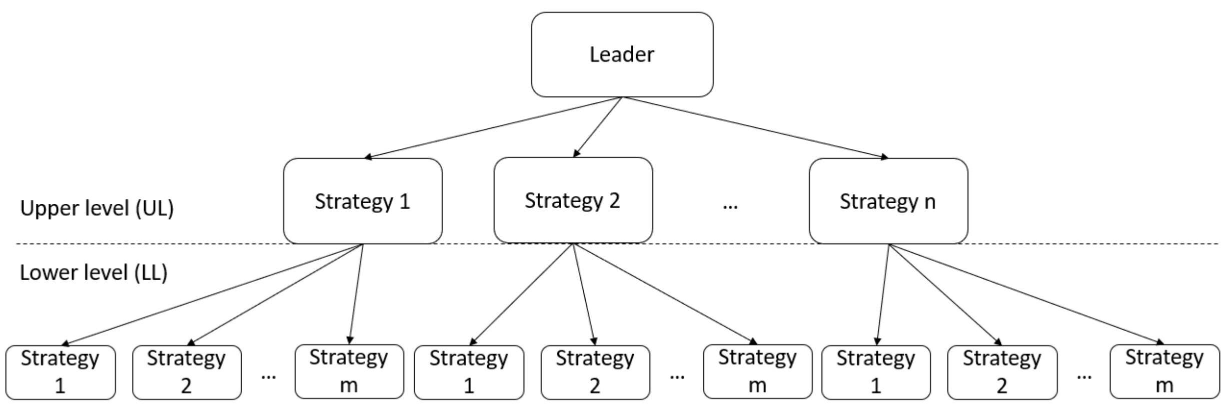


Fig. 3: A bilevel game scheme

The scheme presented in Figure 3 corresponds to a one leader - one follower scenario. In our market model, the lower level consists of individual optimization problems for each agent, which technically falls under the one-leader-multi-follower case. However, we show that these problems are disjoint and thus can be viewed as a single optimization problem at the lower level. Since the optimization models solved over time differ only in parameters, we present a model developed for a single time step and omit the time indexing.

Table 1 provides a summary of the notation for the mathematical formulation that follows.

Symbol	Description
Sets	
$\mathscr{S}$	Set of sensors
$\mathscr{M}$	Set of objects
$\mathscr{K}$	Set of agents
$\mathscr{A}$	Set of arcs in visibility graph
$\mathscr{K}' = \mathscr{K} \cup \{0\}$	Extended set of agents (includes CA)
$\mathscr{S}(k) \subseteq \mathscr{S}$	Set of sensors belonging to agent $k \in \mathscr{K}$
$\mathscr{M}(s) \subseteq \mathscr{S}$	Set of objects visible to sensor $s \in \mathscr{S}$
Parameters	
w_{ij}^k	Interest weight of agent $k \in \mathscr{K}$ for observation $(i,j) \in \mathscr{S} \times \mathscr{M}$
$k(i)$	Owner of sensor $i \in \mathscr{S}$
b_k	Budget for Agent k

Table 1: Notation Reference

4. MATHEMATICAL FORMULATION

In this section, we present our market-based model for a Multi-Agent Sensor Network (MASN), which provides a coordinated sensor assignment (tasking) and maximizes the utility received by all agents in the system. We formulate a bilevel program and show that for the problem at hand relaxing the followers' optimality can benefit the entire system while simplifying the problem.

Optimization variables

- $p_i > 0$ - price of selling sensor $i \in \mathscr{S}$ to another agent
- $s_{ij} \in \{0,1\}$ - assigning sensor $i \in \mathscr{S}$ to object $j \in \mathscr{M}$ (using own sensors); s_{iN+1} denotes selling
- $z_{ij}^k \in \{0,1\}$ - agent $k \in \mathscr{K}$ assigning sensor $i \in \mathscr{S}$ they buy to object $j \in \mathscr{M}$

The model

$$\max_{p,s,z} \sum_{(i,j)\in\mathscr{A}} w_{ij}^{k(i)} s_{i,j} + \sum_i p_i s_{iN+1} + \sum_{k\in\mathscr{K}, i\in\bar{S}(k), j\sim i} (w_{ij}^k - p_i) z_{ij}^k \tag{1a}$$

$$+ \sum_{(i,j)\in\mathscr{A}} w_{ij}^0 (s_{ij} + \sum_{k\in\mathscr{K}\setminus\{k(i)\}} z_{ij}^k) + \sum_{(i,j)\in\mathscr{A}} (w_{ij}^0 - p_i) z_{ij}^0$$

s.t.

$$\sum_{k\in\mathscr{K}'\setminus\{k(i)\}, j\sim i} z_{ij}^k = s_{iN+1} \quad \forall i \in \mathscr{S} \tag{1b}$$

$$\sum_{i\sim j} z_{ij}^0 \leq 1 \quad \forall j \in \mathscr{M} \tag{1c}$$

$$p_i z_{ij}^0 \leq w_{ij}^0 \quad \forall (i,j) \in \mathscr{A} \tag{1d}$$

$$\sum_{(i,j)\in\mathscr{A}} p_i z_{ij}^0 \leq b_0 \tag{1e}$$

$$s_{ij}, z_{ij}^k \in \arg\max_{s_{ij}, z_{ij}^k} \{ \sum_{i\in\mathscr{S}(k), j\sim i} w_{ij}^k s_{i,j} + \sum_{i\in\mathscr{S}(k)} p_i s_{iN+1} + \sum_{i\in\bar{\mathscr{S}}(k), j\sim i} (w_{ij}^k - p_i) z_{ij}^k :$$
$$\sum_{j\sim i} s_{ij} + s_{iN+1} \leq 1 \quad \forall i \in \mathscr{S}(k)$$
$$\sum_{i\in S(k)} s_{ij} + \sum_{i\in\bar{S}(k)} z_{ij}^k \leq 1 \quad \forall j \in \mathscr{M} \tag{1f}$$
$$p_i z_{ij}^k \leq w_{ij}^k \quad \forall i \in \bar{\mathscr{S}}(k), j \in \mathscr{M}$$
$$\sum_{i\in\bar{\mathscr{S}}(k), j\sim i} p_i z_{ij}^k \leq b_k \}$$

$$p > 0, s, z \in \{0, 1\}$$

Upper level decisions

The leader's objective (1a) maximizes the total utility received by all participants (sum of K individual utilities of agents: observing + selling + buying) plus CA's own utility associated with their objects of interest (observed either through CA buying observations, or by another agent). Constraints (1b) ensure that when a sensor is sold, it is sold to exactly one agent and no agent other than the owner uses it if it is not sold. Constraints (1c) make sure that the leader does not observe any object more than once. Constraints (1d) provide an upper bound on a leader's observation purchase, saying that they won't pay more than the utility of that observation. We need them, because in this model, what is paid by one agent is received by another, which creates sets of alternative solutions in terms of price. Constraint (1e) enforces the budget restriction for the leader.

Lower level decisions

The problem for follower k is defined as follows (1f). The objective maximizes their utility, calculated as the utility from using their own sensors, selling observations from their sensors to others and buying observations from others. The first set of constraints ensures that agents use their sensors in exactly one way (including selling). The second constraint set ensures the same for buying sensors. The third constraint set states that agents observe each object no more than once (using either their own sensors or purchased ones). The last constraint restricts the budget for buying sensors from others.

4.1 Pricing considerations

As was mentioned earlier, when a trade occurs, the above model has a continuous range of alternative solutions associated with the price of that trade. It follows from the design of the simplex method used as part of the solution process, that the solution would have each price set to one of the extreme points (ends of the alternative solution range). While mathematically these alternative solutions are equally good, in reality, setting the price of a trade to either 0 or the maximum value a buyer can pay is a questionable pricing policy. Moreover, some considerations of the willingness to sell a sensor can be introduced.

An agent might be willing to sell their sensor for no less than x units, where x is the marginal utility they could get by using it themselves. If one agent is willing to sell their sensor at a price no less than x and there is a buyer k who can pay no more than w_{ij}^k and $x < w_{ij}^k$, the price realizations in range $[x, w_{ij}^k]$ form a set of alternative solutions. One way to look at w_{ij}^k is as the pure utility of an observation for the buyer (what their utility would be if they were given this observation for free). At the same time, it can be treated as the upper bound on how much the buyer is willing to pay for this observation. If they pay that amount, they will get a utility of zero from doing this observation, which makes them indifferent. These are the two extreme cases. For both extremes and everything in between, the total utility of the system will be constant. In that case, it becomes a matter of how the buyer and seller split this utility. One logical way of doing it would be fixing the price to be $\frac{x+w_{ij}^k}{2}$, in which case the utility is split equally. This can be accomplished by introducing constraints of the following form to the upper-level problem.

$$p_i = \frac{\max_k\{w_{ij}^k z_{ij}^k\} + y_i}{2} \quad \forall i \in \mathscr{S},\ k \in \mathscr{K},\ j \in \mathscr{M} \tag{2}$$

where y_i denotes the lowest price the seller would agree to and is determined as follows.

$$y_i \geq w_{ij}^{k(i)}(1 - \sum_{t \in \mathscr{S}(k(i))} s_{tj} - \sum_{r \in \bar{\mathscr{S}}(k(i))} z_{rj}^{k(i)}) \quad \forall i \in \mathscr{S},\ j \in \mathscr{M} \tag{3a}$$

$$y_i \geq (w_{ij}^{k(i)} - w_{tj}^{k(i)})s_{ti} \quad \forall i \in \mathscr{S},\ j \in \mathscr{M},\ t \in \mathscr{S}(k(i)) \tag{3b}$$

$$y_i \geq (w_{ij}^{k(i)} - (w_{rj}^{k(i)} - p_r))z_{rj}^{k(i)} \quad \forall i \in \mathscr{S},\ j \in \mathscr{M},\ r \in \bar{\mathscr{S}}(k(i)) \tag{3c}$$

Constraint set (3a) states that the price should be no less than the utility the owner gets from making an observation with this sensor provided that they do not observe that object with any other sensor. Constraints (3b) ensure that the price is no less than the owner's utility of using this sensor to observe an object currently observed by another sensor they own (replacing own observation). In a similar manner, constraints (3c) ensure that the price is no less than the utility of replacing a purchased observation with an observation using that sensor.

Constraint (2) contains a max operator and cannot be introduced in a mixed integer program (MIP) as is. However, the lower bound can be obtained as:

$$p_i \geq \frac{w_{ij}^k z_{ij}^k + y_i}{2} \quad \forall i \in \mathscr{S},\ k \in \mathscr{K},\ j \in \mathscr{M},\ l \in \mathscr{M} \tag{4}$$

The upper bound can then be enforced by adding a regularization term $-\varepsilon \sum_{i,j,k} p_i s_{iN+1}$ to the objective given a small ε.

This approach helps us to break the symmetry and ensures a realistic price setting. However, introducing this in a bilevel problem may create feasibility issues. Let us illustrate one such issue with a small example. Consider an instance with three agents and a sensor owned by Agent 1. For simplicity, let us assume that all the other sensors are unavailable due to visibility constraints. The weights for observing an object with that sensor are: Agent 1 - 3, Agent 2 - 5, and Agent 3 - 6. In this case, selling that observation to Agent 3 will maximize the total utility. According to the "fair price" requirement described above, the price of this transaction should be 4.5. Given a fixed price of 4.5, both Agent 2 and Agent 3 would be willing to buy that observation, while in reality it can only be sold to one agent. In the bilevel model adopted here, the followers do not see the upper level constraint (2) and the solution in this case does not exist because the optimality of both Agent 1 and Agent 2 cannot be respected simultaneously so the model will be forced to choose the alternative solution where the price is equal to 5. In this case, in a real system, we would be willing to sacrifice the optimality of one of the agents in order to get a fair solution.

The above example can justify going away from a bilevel approach and using a single-level optimization model, where the CA matches the buyers and the sellers and fixes the price at the midpoint.

4.2 Relaxed single-level model

Here we present a market-based MASN model with single-level optimization, where we removed the optimality requirement for each agent and introduced additional constraints to make sure that an agent will never be forced to commit to an observation that does not add to their utility function. The resulting model consists of the upper-level problem described earlier and all the constraints from individual lower-level problems.

$$\max_{p,s,z} \sum_{(i,j)\in\mathscr{A}} w_{ij}^{k(i)} s_{i,j} + \sum_i p_i s_{iN+1} + \sum_{k\in\mathscr{K},i\in\bar{S}(k),j\sim i} (w_{ij}^k - p_i) z_{ij}^k + \sum_{(i,j)\in\mathscr{A}} w_{ij}^0 (s_{ij} + \sum_{k\in\mathscr{K}\setminus\{k(i)\}} z_{ij}^k) + \sum_{(i,j)\in\mathscr{A}} (w_{ij}^0 - p_i) z_{ij}^0 - \varepsilon \sum_{i,j,k} p_i s_{iN+1} \tag{5a}$$

s.t.

$$\sum_{k\in\mathscr{K}'\setminus\{k(i)\},j\sim i} z_{ij}^k = s_{iN+1} \quad \forall i \in \mathscr{S} \tag{5b}$$

$$\sum_{i\sim j} z_{ij}^0 \leq 1 \quad \forall j \in \mathscr{M} \tag{5c}$$

$$p_i z_{ij}^0 \leq w_{ij}^0 \quad \forall (i,j) \in \mathscr{A} \tag{5d}$$

$$\sum_{(i,j)\in\mathscr{A}} p_i z_{ij}^0 \le b_0 \tag{5e}$$

$$\sum_{j\sim i} s_{ij} + s_{iN+1} \le 1 \quad \forall i \in \mathscr{S} \tag{5f}$$

$$\sum_{j\sim i} z_{ij}^k \le 1 \quad \forall i \in \bar{\mathscr{S}}(k), k \in \mathscr{K} \tag{5g}$$

$$\sum_{i\in S(k)} s_{ij} + \sum_{i\in \bar{S}(k)} z_{ij}^k \le 1 \quad \forall j \in \mathscr{M} \tag{5h}$$

$$p_i z_{ij}^k \le w_{ij}^k \quad \forall i \in \bar{\mathscr{S}}(k), j \in \mathscr{M}, k \in \mathscr{K} \tag{5i}$$

$$\sum_{i\in\bar{\mathscr{S}}(k), j\sim i} p_i z_{ij}^k \le b_k \quad \forall k \in \mathscr{K} \tag{5j}$$

$$p_i \ge \frac{w_{ij}^k z_{ij}^k + y_i}{2} \quad \forall i \in \mathscr{S},\ k \in \mathscr{K},\ j \in \mathscr{M},\ l \in \mathscr{M} \tag{5k}$$

$$y_i \ge w_{ij}^{k(i)}\Big(1 - \sum_{t\in\mathscr{S}(k(i))} s_{tj} - \sum_{r\in\bar{\mathscr{S}}(k(i))} z_{rj}^{k(i)}\Big) \quad \forall i \in \mathscr{S},\ j \in \mathscr{M} \tag{5l}$$

$$y_i \ge (w_{ij}^{k(i)} - w_{tj}^{k(i)}) s_{ti} \quad \forall i \in \mathscr{S},\ j \in \mathscr{M},\ t \in \mathscr{S}(k(i)) \tag{5m}$$

$$y_i \ge (w_{ij}^{k(i)} - (w_{rj}^{k(i)} - p_r)) z_{rj}^{k(i)} \quad \forall i \in \mathscr{S},\ j \in \mathscr{M},\ r \in \bar{\mathscr{S}}(k(i)) \tag{5n}$$

$$p > 0, s, z \in \{0,1\}$$

This model allows for conflicts arising when multiple agents are interested in buying the same sensor for a given price to be resolved in favour of the agent with the highest utility. We use this model and a heuristic solution derived for it to demonstrate the proposed collaboration mechanism in Section 6.

5. HEURISTIC SOLUTION

After applying some classical linearizations (see [6, 7, 15, 25]), formulation (5) presents a MIP and is NP-hard, and therefore, can be impractical for large instances under realistic time requirements. In this section, we present a greedy algorithm that provides a suboptimal solution but is computationally efficient and might be helpful in practice.

The basic idea of our greedy algorithm is to iteratively make assignments with the largest marginal utility and block all the conflicting assignments. Figure 4 features a block diagram explaining the algorithm.

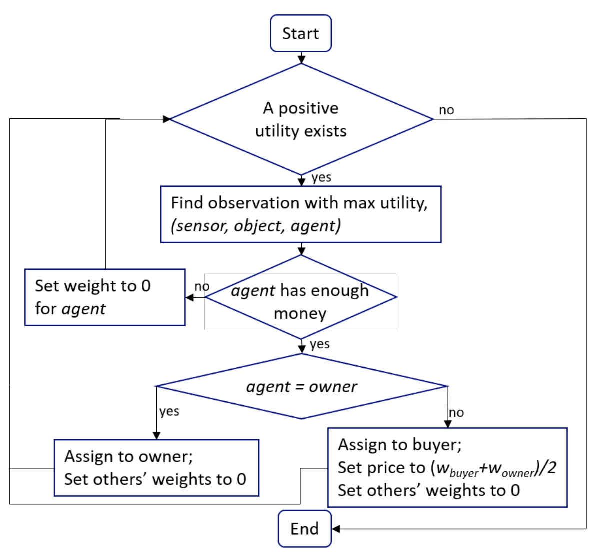


Fig. 4: Block diagram for the greedy heuristic

The greedy heuristic runs in $O(MNK(log(MNK)+d_{max}))$ where K is the number of agents, N is the number of sensors, M is the number of objects to be observed and d_{max} is the maximum degree in the bipartite sensor-object visibility graph. A detailed pseudocode is presented below.

```
Data: w_ij^k - observations utilities, owner - sensor ownership
b_k - agents budget
while Σ_{i,j,k} w_ij^k > 0 do
    sensor, object, agent = arg max_{i,j,k} {w_ij^k}
    if agent = owner[sensor] then
        assignOwn(sensor, object)
    else
        if b_agent ≥ w_{sensor,object}^{agent} then
            assignBought(sensor, owner, agent)
            b_agent −= (w_{sensor,object}^{agent} + w_{sensor,object}^{owner[sensor]}) / 2
            for k = 0..K do
                if k ≠ agent then
                    w_{sensor,object}^k = 0
                end
            end
        else
            w_{sensor,object}^{agent} = 0
        end
    end
end
```

Algorithm 1: Sensor assignment based on greedy heuristic

This approach provides a substantial reduction in computation time compared to the MIP formulation presented earlier. It allows us to solve a realistic instance with 290 sensors, 22,028 objects, 23 agents and 134,286 visibility connections in 22 seconds instead of more than 8 hours to compute the exact solution (the calculations were performed on a 2018 MacBook Pro with 2.7 GHz Quad-Core Intel Core i7 and 16 GB RAM). The results on the scalability of both the heuristic approach and the exact formulation are summarized in Table 2.

Instance			Exact			Greedy		
N	M	A	Time, s	Objective	Gap, %	Time, s	Objective	Gap, %
87	100	100	<1	24.98	0	<1	24.98	0.00
279	943	1,000	3	129.47	0	<1	129.46	0.01
290	1,761	2,000	7	171.41	0	<1	171.40	0.01
290	2,524	3,000	17	183.21	0	<1	183.20	0.01
290	3,819	5,000	76	201.75	0	<1	201.73	0.01
290	6,208	10,000	419	220.15	0	<1	220.12	0.01
290	14,017	50,000	1,246	251.88	0	4	251.82	0.02
290	18,041	100,000	43,347			13	262.14	
290	20,831	200,000				47	268.54	
290	22,028	521,337				211	275.02	

Table 2: Routing costs reduction, 1 hour

6. PRELIMINARY SIMULATION RESULTS

In this section we present preliminary simulations of distributed sensor networks tracking a large population of space objects and use this modeling environment to analyze the utility of the market approach for coordinating observations.

The simulation starts with the orbital parameters for more than 22,000 space objects in the public catalog from Space-Track.org given as two-line element sets (TLEs). Space dynamics calculations and related tasks such as orbit propagation, orbit determination, or illumination and viewing geometry are implemented using the Orekit library (Python API). TLEs from the catalog are propagated forward in time and are taken to represent the "true" motion of each object.

The model also includes networks of ground-based sensors distributed around the world and controlled by a set of autonomous agents. Each sensor is given a fixed geographic location and can be configured to collect electro-optical (EO) or radio-frequency radar (RF) observations with user-selected accuracy. The main difference is that the RF sensors in the simulation can measure distance, while the EO sensors deliver angles-only data. Each agent is responsible for managing a subset of all sensors, primarily for deciding which sensor will observe which target object in a given time step. Agents also interact with the coordination authority to establish sensor pricing, as well as, buy and sell resources based on those prices. Another functionality provided by agents is to receive all simulated observations from sensors and update orbit estimates for relevant objects. Orbit estimation is performed using a Kalman filter for each space object with an underlying estimated TLE that changes from one time step to the next relative the assumed "true" trajectory given by the catalog TLE.

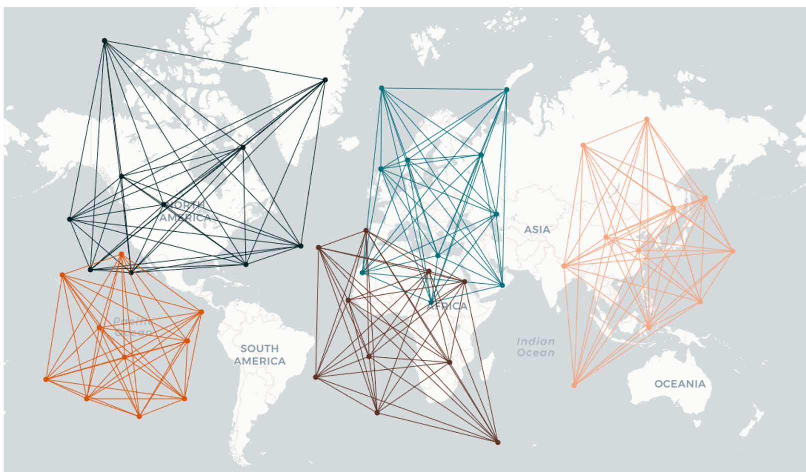


Fig. 5: Five examples of hypothetical regional sensor subnetworks in the simulation (out of 23 total).

Configuring a sizeable globally distributed sensor network takes a significant amount of user input. For our analysis we configured a set of 23 agents managing 290 sensors, between one and two dozen sensors per agent. The distribution of geographic locations of sensors corresponds roughly to a set of regional networks with each regional subnetwork managed by a single agent and locations of specific sensors chosen randomly (Figure 5). Together, those 23 subnetworks span the full range of geographic longitudes and latitudes that sample a large diversity of illumination and viewing conditions around the world at any given time. Each agent managing a single subnetwork is given priority weights for all space objects in the simulation.

For this analysis we used diversified agent preferences across different orbital regimes with some agents that prioritize LEO objects, some that focus on GEO, and some on MEO. We also included agents that strongly prioritize just a few dozen objects, as well as, agents with a broad distribution of weights and no strong preferences. All sensors in this preliminary simulation were EO sensors capable of localizing an object in a single time step down to 0.2 arcseconds.

A single time step in the rolling horizon optimization corresponds roughly to the amount of time necessary for executing a single observation by a single sensor, which may typically take about 1 minute (Figure 6). In each optimization interval the positions and velocities of all space objects in the simulation are propagated to a new time step. This information is then used to compute the apparent positions of all objects in the local sky coordinates for each sensor. Based on this information, the simulation determines the visibility graph with arcs pointing from each potential observer to any potential target. For optical sensors this requires that the target object is illuminated by the Sun and above the local horizon for an observer on the night time side of the Earth. Simulated observations are carried out at the end of each optimization period according to the optimized sensor assignment from Algorithm 1.

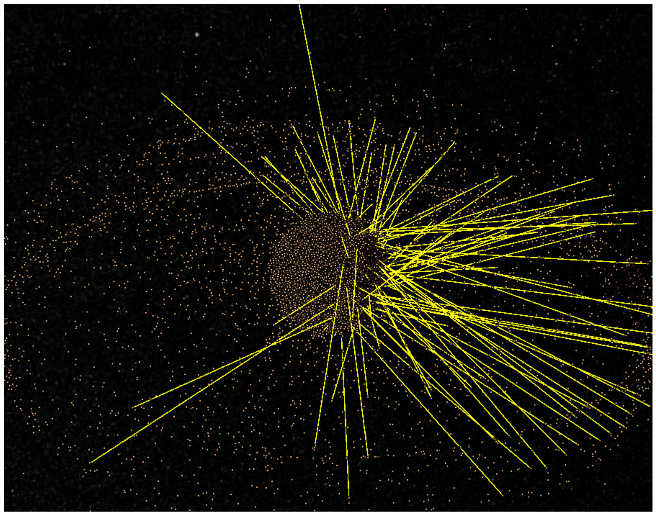


Fig. 6: Visualization of sensor assignment for a single time step of the rolling horizon optimization running in our simulated Multi-Agent Sensor Network.

The simulation tracks aggregate utilities (weights) for agents and individual sensor-object pairs. Each weight is a product of the object priority for a given agent and the intrinsic value associated with the object. In the context of Space Traffic Management, the intrinsic value of an object can be linked to collision risk, which generally increases with increasing localization uncertainties and proximity to other objects. The localization uncertainty (position covariance) tends to grow with time since the last observation, providing a strong incentive to minimize gaps between observations of high priority objects.

Figure 7 shows a comparison between coordinated and uncoordinated observing scenarios. In a coordinated multi-agent sensor network, the CA is running the observation exchange allowing agents to trade sensor time. In an uncoordinated network there is no trading and each agent assigns its own sensors to maximizes its own utility. Under these assumptions, trading observations allows the agents to almost double their utility.

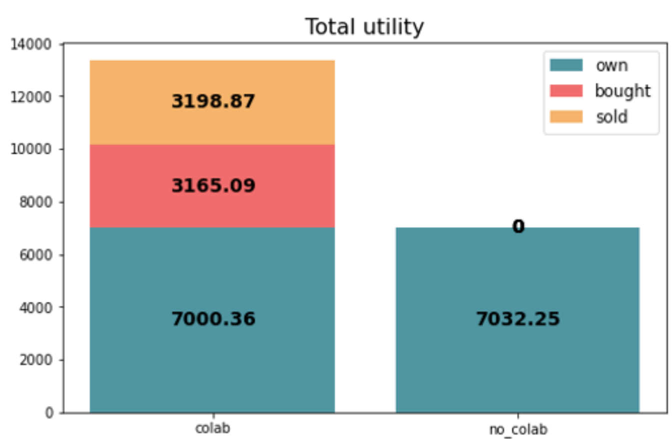


Fig. 7: Total utility of all agents in a coordinated multi-agent network with trading of observations (left) compared to an uncoordinated network with no trade (right).

7. FUTURE WORK

While our preliminary results indicate that a market-based model proposed here may provide an effective coordination of multi-agent sensor networks for SDA, the full potential of this approach is still to be understood. A straightforward continuation of this work is to add more realism to the simulation, explore performance gains for different types of sensors, geographic distributions, and object importance gains, and study the impact of different object importance weights and different ways to calculated risk. Another interesting problem is the long-term evolution of risk in the context of Space Traffic Management and collision avoidance. We are also planning to investigate the influence of budget constraints on the optimal sensor tasking and the possibility of reducing computation time.

8. REFERENCES

[1] Jesús Cerquides, Alessandro Farinelli, Pedro Meseguer, and Sarvapali D. Ramchurn. A tutorial on optimisation for multi-agent systems. 2013.

[2] T. Holvoet D. Weyns, N. Boucke and B. Demarsin. Dyncnet: A protocol for dynamic task assignment in multiagent systems. *First International Conference on Self-Adaptive and Self-Organizing Systems (SASO 2007)*, pages 281–284, 2007.

[3] Rajdeep K. Dash, Nicholas R. Jennings, and David C. Parkes. Computational-mechanism design: A call to arms. *IEEE Intelligent Systems*, 18(6):40–47, nov 2003.

[4] Giancarlo Fortino. Agents meet the iot: Toward ecosystems of networked smart objects. *IEEE Systems, Man, and Cybernetics Magazine*, 2(2):43–47, 2016.

[5] Carolin Frueh, Hauke Fielder, and Johannes Herzog. Heuristic and optimized sensor tasking observation strategies with exemplification for geosynchronous objects. *Journal of Guidance, Control, and Dynamics*, 41(5):1036–1048, 2018.

[6] Fred Glover. Improved linear integer programming formulations of nonlinear integer problems. *Management science*, 22(4):455–460, 1975.

[7] Fred Glover and Eugene Woolsey. Converting the 0-1 polynomial programming problem to a 0-1 linear program. *Operations research*, 22(1):180–182, 1974.

[8] Keric Hill, Paul Sydney, Kris Hamada, Randy Cortez, Kim Luu, Moriba Jah, Paul W Schumacher, Michael Coulman, Jeff Houchard, and Dale Naho'olewa. Covariance-based network tasking of optical sensors. *Advances in the Astronautical Sciences*, 136:769–786, 2010.

[9] Andreas Hinze, Hauke Fiedler, and Thomas Schildknecht. Optimal scheduling for geosynchronous space object follow-up observations using a genetic algorithm. In *Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS)*. Maui Economic Development Board Maui, HW, 2016.

[10] D Jang, PM Siew, G Gondelach, and R Linares. Space situational awareness tasking for narrow field of view sensors: A deep reinforcement learning approach. In *71st International Astronautical Congress. International Astronautical Federation, the International Academy of Astronautics, and the International Institute of Space Law, Virtual*, 2020.

[11] Andris D Jaunzemis, Marcus J Holzinger, and Moriba K Jah. Evidence-based sensor tasking for space domain awareness. In *Advanced Maui Optical and Space Surveillance Technologies Conference*, page 33. Maui Economic Development Board Maui, HI, 2016.

[12] Richard Linares and Roberto Furfaro. Dynamic sensor tasking for space situational awareness via reinforcement learning. In *Advanced Maui Optical and Space Surveillance Technologies Conference*, page 36. Maui Economic Development Board Maui, HI, 2016.

[13] Richard Linares and Roberto Furfaro. An autonomous sensor tasking approach for large scale space object cataloging. In *Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS)*, pages 1–17, 2017.

[14] Bryan D Little and Carolin E Frueh. Space situational awareness sensor tasking: comparison of machine learning with classical optimization methods. *Journal of Guidance, Control, and Dynamics*, 43(2):262–273, 2020.

[15] Garth P McCormick. Computability of global solutions to factorable nonconvex programs: Part i—convex underestimating problems. *Mathematical programming*, 10(1):147–175, 1976.

[16] R Musci, T Schildknecht, M Ploner, and G Beutler. Orbit improvement for gto objects using follow-up observations. *Advances in Space Research*, 35(7):1236–1242, 2005.

[17] Kevin M Nastasi and Jonathan T Black. Adaptively tracking maneuvering spacecraft with a globally distributed, diversely populated surveillance network. *Journal of Guidance, Control, and Dynamics*, 42(5):1033–1048, 2019.

[18] Simon Parsons, Juan A. Rodriguez-Aguilar, and Mark Klein. Auctions and bidding: A guide for computer scientists. *ACM Comput. Surv.*, 43(2), feb 2011.

[19] Pablo Pico-Valencia and Juan Antonio Holgado-Terriza. Agentification of the internet of things: A systematic literature review. *International Journal of Distributed Sensor Networks*, 14, 2018.

[20] Peng Mun Siew, Daniel Jang, and Richard Linares. Sensor tasking for space situational awareness using deep reinforcement learning. In *Proceedings of the AAS/AIAA Astrodynamics Specialist Conference, Big Sky, MT, USA*, pages 9–11, 2021.

[21] Peng Mun Siew, Daniel Jang, Thomas G Roberts, and Richard Linares. Space-based sensor tasking using deep reinforcement learning. *The Journal of the Astronautical Sciences*, pages 1–38, 2022.

[22] Reid G. Smith. The contract net protocol: High-level communication and control in a distributed problem solver. *IEEE Transactions on Computers*, C-29(12):1104–1113, 1980.

[23] Reid G. Smith and Randall Davis. Frameworks for cooperation in distributed problem solving. *IEEE Transactions on Systems, Man, and Cybernetics*, 11(1):61–70, 1981.

[24] Heinrich Von Stackelberg. *Marktform und gleichgewicht*. J. springer, 1934.

[25] Lawrence J Watters. Reduction of integer polynomial programming problems to zero-one linear programming problems. *Operations Research*, 15(6):1171–1174, 1967.

[26] Patrick S Williams, David B Spencer, and Richard S Erwin. Coupling of estimation and sensor tasking applied to satellite tracking. *Journal of Guidance, Control, and Dynamics*, 36(4):993–1007, 2013.

[27] Guohua Wu, Qizhang Luo, Xiao Du, Yingguo Chen, Ponnuthurai Nagaratnam Suganthan, and Xinwei Wang. Ensemble of metaheuristic and exact algorithm based on the divide-and-conquer framework for multisatellite observation scheduling. *IEEE Transactions on Aerospace and Electronic Systems*, 58(5):4396–4408, 2022.