Received: 25 October 2021 | Revised: 5 September 2022 | Accepted: 11 November 2022

DOI: 10.1002/rnc.6510

RESEARCH ARTICLE

WILEY

An exact solution approach for the mobile multi-agent sensing problem

Gwang Kim[1] | **Jongmin Lee**[2] | **Ilkyeong Moon**[2,3]

[1]Division of Business Administration, Chosun University, Gwangju, Korea

[2]Department of Industrial Engineering, Seoul National University, Seoul, Korea

[3]Institute of Engineering Research, Seoul National University, Seoul, Korea

Correspondence
Ilkyeong Moon, Institute of Engineering Research, Seoul National University, Seoul, Korea.
Email: ikmoon@snu.ac.kr

Funding information
National Research Foundation of Korea (NRF), Grant/Award Number: NRF-2021R1A4A1029124, RS-2022-00165977

Abstract

Multi-agent systems are generally applicable in various fields and aim to coordinate the decisions from agents, each of which makes a local decision. In particular, research on operations management with mobile multi-agents such as drones has gained prominence in recent years. In this article, we present a mobile multi-agent sensing problem, which is formulated as a submodular maximization problem under a partition matroid constraint. When events detected by agents lead to severe and catastrophic consequences, obtaining (near)-optimal solutions by using exact algorithms is crucial to reducing the probability of harmful situations. Therefore, in this article we propose an exact solution approach, using two valid inequalities for our problem to find a (near)-optimal solution. Moreover, we deal with the risk-averse decision and its cutting-plane algorithm for our problem, which is to maximize *conditional value-at-risk* (CVaR). Finally, we show the performance of our algorithms through numerical experiments, including a case study on forest fires.

KEYWORDS

conditional value-at-risk, cutting-plane algorithm, mobile multi-agent systems, submodular functions, valid inequality

1 | INTRODUCTION

A multi-agent system refers to the computerized system composed of multiple interacting intelligent agents and is known as a suitable tool for data processing, data control, and data expertise. The objective of the system is to coordinate the decisions from all agents, which each makes a local decision, in order to optimize the sum of the agents' local objective function.[1-3] These systems are generally applicable in various fields, such as air traffic management,[4] transportation systems,[5,6] and structural health monitoring.[7] Recently, the systems have been expanded to cover sensing problems, especially in sensor fault estimation,[8] remote-sensing images,[9] detection of forest fires,[10,11] and energy management.[12]

If agents in a system are movable, we have to consider the system as a mobile multi-agent system. Mobile multi-agents that can move and interact with each other are widely applicable in diverse domains, such as traffic control and management systems,[13] the NASA sensor web,[14] and collision avoidance.[15] When it comes to the mobile multi-agent system, there has been interest in designing a practical model with mobile agents and enhancing the flexibility of the model. Chen et al. proposed a mobile multi-agent system that improves the flexibility and adaptability of large traffic management systems through simulation results.[13] Pavone et al. studied a dynamic vehicle routing system for mobile robotic networks and wireless ad hoc networks in mobile multi-agent systems.[16] Su et al. presented a second-order consensus problem with mobile multi-agents that have a characteristic of nonlinear dynamics.[17] Gu et al. considered a cooperative detection

problem in mobile multi-agent networks.[18] They emphasized the importance of introducing mobile multi-agents in the problem because they enhance flexibility and accessibility in large networks.

This study focuses on a sensing problem by utilizing mobile multi-agents as sensors. If agents are movable, then effective and efficient operations management would be possible in terms of event detection, sensing, and monitoring.[18-20] The characteristics of the sensor as a mobile agent are classified into two types: deterministic sensing characteristics and probabilistic sensing characteristics.[21] Between the two types, the probabilistic sensing model has been used more in research on mobile multi-agent systems.[19,22] This model is more reasonable because the sensing probability distribution is determined according to the distance between the agent and the target. Our model also reflects the realistic situation by exploiting the Elfes sensing model, which is known as one of the probabilistic sensing models and has widely been applied in recent research.[23,24]

Using a given set of mobile multi-agents, we present a problem in which the objective is to maximize the sum of detection probabilities from the nodes in one period. It is assumed that the probability of event occurrence at each node is not constant and changes irregularly, regardless of periods. This feature can be applied in detecting forest fires, uncertain traffic incidents, or oil spillage in a large area.[13,25,26] Agents execute an event detection task at a point and then move to another point based on current probabilities of event occurrence at nodes. It is more likely to get solutions that have more prediction accuracy when the solutions are obtained by utilizing immediately updated probabilities, not those in the old times. For this reason, the agents repeat the process in which the problem is solved given the changeable probabilities in every period.

The sum of detection probabilities, the objective function of our problem, has the properties of monotone increasing and submodular. A feature of submodularity refers to the marginal gain diminishing. Sensing problems with these properties have been studied in various research fields.[27-30] Shamaiah et al. presented a sensor selection problem in resource-constrained sensor networks. They covered the problem as the maximization of a submodular function over uniform matroids represented as selecting a subset of the given sensors.[30] Jawaid and Smith considered a sensor scheduling problem to minimize the estimation error at a terminal time under linear dynamical systems.[28] Sun et al. studied a coverage problem with stationary multi-agents to maximize a joint event detection probability, which is submodular under a uniform matroid.[31] Contrary to these studies, this article expands to a sensing problem under a partition matroid because it is appropriate for using mobile multi-agent systems. That is, each movable agent has to choose only one position (strategy) among all positions available. The problem is known to be NP-hard.[32,33] Effective and efficient solution methodologies are needed to solve the sensing problem.

There has been previous literature on presenting approximation algorithms to deal with the sensing problems inherent in submodularity. Approximation ratios that compare solutions obtained by algorithms with optimal solutions are generally adopted to measure performance.[34] Previous literature has presented algorithms based on the greedy approach by using the concept of an approximation ratio.[35-37] Shamaiah et al. presented a sensor selection problem under uniform matroids and proved that a greedy sensor selection algorithm achieved performance within $1 - \frac{1}{e}$ of the optimal solution.[30] Moreover, instance-dependent guarantees of the greedy-type algorithm have been presented in the sensing problems with submodularity.[20,38-40] (Instance-dependent guarantees are guarantees in which the approximation ratio is dependent on the instances.) However, when events detected by multi-agents lead to severe or catastrophic consequences, obtaining (near)-optimal solutions by using exact algorithms might be more critical to reducing the probability of harmful situations.

In this article, we design an exact algorithm for the mobile multi-agent sensing problem to find a (near)-optimal solution. There have not been many studies on exact algorithms to obtain (near)-optimal solution for sensing problems with submodularity. Nemhauser and Wolsey and Ko et al. presented branch-and-bound algorithms for maximizing a submodular set function under a uniform matroid.[41,42] Ko et al. showed the computational results through numerical experiments, but only analyzed the problem situation in which the number of decision variables is less than 40.[42] Paulson and Mesbah used the sample average approximation for stochastic optimal control with joint chance constraints.[43] Kawahara et al. presented a cutting-plane algorithm for the sensor placement problem.[44] The objective was to maximize a submodular set function under a cardinality constraint, which is identical to a uniform matroid. However, little research has been conducted to design an exact algorithm that considers a partition matroid constraint. Numerous decision variables are likely to be needed to formulate the problem with a partition matroid constraint, and the number of feasible solutions becomes enormous. For these reasons, it is important to present an exact algorithm in terms of efficiency. This article introduces a cutting-plane algorithm using two valid inequalities to solve the problem. We derive two valid inequalities (optimality cuts), which are based on the propositions of Nemhauser et al.,[33] to achieve feasibility and to find a (near)-optimal solution efficiently. In our algorithm, we add these valid inequalities in every iteration. We also present how effective these valid inequalities are through numerical experiments.

Moreover, in practice, there is a possibility of failure in sensing targets, due to errors or problems with sensors. Our model also attempts to solve the mobile multi-agent sensing problem by considering such uncertain situations. Previous literature reflected uncertainty in problems by introducing an expected value of the objective function with a random variable.[45,46] If the data were uncertain and of high variability, the solution obtained by the expected function might lead to a poor performance with a high probability.[46] In this article, we also deal with the mobile multi-agent sensing problem with a risk-averse measure. We use *conditional value-at-risk* (CVaR), which is known as a popular risk-averse measure and defined as the expectation of the worst α-tail scenarios.[47]

To deal with uncertain situations, there has been research on maximizing CVaR in sensing problems. Maehara showed that CVaR is not necessarily submodular even if the objective function is monotone submodular.[45] An auxiliary function is introduced to make the function submodular and shows the equivalent results when optimizing the function and CVaR. Zhou and Tokekar considered that there is uncertainty in the value of the objective function because sensors might fail.[47] They proposed a sequential greedy algorithm for the sensing problems using CVaR. Wu et al. studied a set covering problem in which the objective was to maximize CVaR, which is equivalent to maximizing a submodular set function under a uniform matroid.[48] Contrary to the previous research, we cover a risk-averse problem for maximizing a submodular set function under a partition matroid. The problem under a partition matroid tends to have a greater number of decision variables than a uniform matroid has. For convergence, we propose a two-stage stochastic programming model based on a sample average approximation. We also present a cutting-plane algorithm for the mobile multi-agent sensing problem considering the risk-averse measure. Similar to the mobile multi-agent sensing problem without uncertain data, we derive two valid inequalities and present the effectiveness of these valid inequalities through numerical experiments.

Our goal in this article is to mathematically present a mobile multi-agent sensing problem to maximize the sum of detection probabilities from the nodes in one period. The problem is NP-hard because it is represented as a submodular maximization problem under a partition matroid constraint. To cope with the difficulty, we design an exact solution approach, using two valid inequalities, to find a (near)-optimal solution efficiently. Another contribution of the article is to deal with the risk-averse decision and its exact algorithm for the problem under uncertain situations (errors with sensors). We show the validity and excellence of the solution approaches theoretically and experimentally.

The remainder of the article is organized as follows. Section 2 defines the mobile multi-agent sensing problem and shows that the problem is formulated as a submodular maximization problem under a partition matroid constraint. In Section 3, we present a cutting-plane algorithm using two valid inequalities to find a (near)-optimal solution efficiently. Section 4 covers the risk-averse problem by using CVaR. We present a cutting-plane algorithm using two valid inequalities to find effective risk-averse solutions. Section 5 provides numerical experiments with algorithms for the two problems (without and with uncertainty). We analyze the performance of the algorithms by considering three methods of choosing initial solutions. A case study is also conducted, based on the detection of forest fires, to show the validity and applicability of the algorithms. We conclude the article in Section 6.

2 | PROBLEM STATEMENT

In this section, a mobile multi-agent sensing problem is defined. Table 1 shows the notations of the problem.

Let $X = \{1, 2, \dots, N\}$ be a set of agents, and $Y = \{1, 2, \dots, M\}$ be a set of nodes. Each agent is movable on a given space $\Omega \subset \mathbb{R}^2$ and it can monitor a set of nodes whose positions are fixed. We then decide the next locations of the agents based on the current locations of the agents. The agent can move to l_i during a unit period. This means that the distance between cl_i and l_i is less than or equal to L_i. In other words, we define S_i as $\{(i, l_i) | \|l_i - cl_i\| \le L_i\}$ in a period. Let $S = S_1 \cup S_2 \cup \cdots \cup S_N$. We simplified the Elfes sensing model into a suitable form and used it as a sensing technique.[23] If $\|l_i - o_j\| \le \delta_i$, $p(s,j)$ becomes $\exp\left(-\lambda_i \|l_i - o_j\|\right)$; otherwise, $p(s,j)$ becomes 0. When it comes to the probability from each node, we calculate $P_j\left(\overline{S}\right)$ as $1 - \prod_{s \in \overline{S}}(1 - p(s,j))$. It is assumed that the detection probabilities of different agents are independent.

The mobile multi-agent sensing problem is expressed as follows:

$$\max_{\overline{S}} \sum_{j \in Y} E_j \times P_j(\overline{S}), \tag{1}$$

$$\text{subject to } |\overline{S} \cap S_i| \le 1, \forall i \in X, \tag{2}$$

TABLE 1 Notations

Notations	Definitions
i	Index of agent ($i \in X$)
j	Index of node ($j \in Y$)
o_j	Position of node j ($o_j \in \Omega$)
cl_i	Current location of agent i
L_i	Maximum distance for agent i in one period
S_i	Set of strategies for agent i
δ_i	Bounded sensing radius of agent i
λ_i	Sensing decay factor of agent i
$p(s,j)$	Probability that agent i detects an event occurrence at node j under a strategy $s = (i, l_i)$
$P_j(\overline{S})$	Joint probability of node j under a strategy set $\overline{S}(\subseteq \mathcal{S})$
E_j	Probability of event occurrence in node j

where node j has the probability of event occurrences E_j. We show that the problem is a submodular maximization problem under a partition matroid constraint. First, Constraint (2) is represented as a condition for *partition matroid*.[33] When we choose a strategy set that includes, at most, one strategy from each disjoint set $\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_N$, it satisfies feasibility. Second, the objective function in this problem is a monotone increasing and submodular set function, which is shown in Theorem 1.

Theorem 1. *The objective function (1) is a monotone increasing and submodular set function.*

Proof. First of all, Given a ground set S, a set function $f : 2^S \to \mathbb{R}$ is defined to be monotone (increasing) if for any $A \subset B \subseteq S, f(A) \leq f(B)$, and submodular if for any $A \subset B \subseteq S$ and $s \notin B, f(B \cup \{s\}) - f(B) \leq f(A \cup \{s\}) - f(A)$.

Consider two strategy sets $\overline{S}_1$ and $\overline{S}_2$ such that $\overline{S}_1 \subseteq \overline{S}_2 \subseteq \overline{S}$. We know that $P_j(\overline{S}_1) \leq P_j(\overline{S}_2)\ \forall j \in Y$ by the definition of P_j. We have $\sum_{j\in Y} E_j \times P_j(\overline{S}_1) \leq \sum_{j\in Y} E_j \times P_j(\overline{S}_2)$. It means that the objective function is monotone increasing. For $s \in \mathcal{S}$ and $s \notin \overline{S}_2$,

$$\begin{aligned}
\sum_{j\in Y} E_j \times P_j\left(\overline{S}_1 \cup \{s\}\right) &= \sum_{j\in Y} E_j \times \left(1 - \prod_{k\in\overline{S}_1\cup\{s\}} \left(1 - p(k,j)\right)\right) \\
&= \sum_{j\in Y} E_j \times \left(1 - \left(1 - p(s,j)\right)\prod_{k\in\overline{S}_1} \left(1 - p(k,j)\right)\right) \\
&= \sum_{j\in Y} E_j \times \left(1 - \prod_{k\in\overline{S}_1} (1 - p(k,j))\right) + \sum_{j\in Y} E_j \times p(s,j) \times \prod_{k\in\overline{S}_1} (1 - p(k,j)) \\
&= \sum_{j\in Y} E_j \times P_j\left(\overline{S}_1\right) + \sum_{j\in Y} E_j \times p(s,j) \times \prod_{k\in\overline{S}_1} \left(1 - p(k,j)\right).
\end{aligned}$$

Above the equations, we know that $\sum_{j\in Y} E_j \times P_j\left(\overline{S}_1 \cup \{s\}\right) - \sum_{j\in Y} E_j \times P_j\left(\overline{S}_1\right)$ becomes $\sum_{j\in Y} E_j \times p(s,j) \times \prod_{k\in\overline{S}_1}(1 - p(k,j))$. In a similar way, $\sum_{j\in Y} E_j \times P_j\left(\overline{S}_2 \cup \{s\}\right) - \sum_{j\in Y} E_j \times P_j\left(\overline{S}_2\right)$ becomes $\sum_{j\in Y} E_j \times p(s,j) \times \prod_{k\in\overline{S}_2}(1 - p(k,j))$. We know that $P_j\left(\overline{S}_1\right) \leq P_j\left(\overline{S}_2\right)\ \forall j \in Y$. It means that $1 - P_j\left(\overline{S}_1\right) \geq 1 - P_j\left(\overline{S}_2\right)\ \forall j \in Y$, which is equal to $\prod_{k\in\overline{S}_1}(1 - p(k,j)) \geq \prod_{k\in\overline{S}_2}(1 - p(k,j))\ \forall j \in Y$. Therefore, $\sum_{j\in Y} E_j \times P_j\left(\overline{S}_1 \cup \{s\}\right) - \sum_{j\in Y} E_j \times P_j\left(\overline{S}_1\right) \geq \sum_{j\in Y} E_j \times P_j\left(\overline{S}_2 \cup \{s\}\right) - \sum_{j\in Y} E_j \times P_j\left(\overline{S}_2\right)$. The objective function is submodular set function. ∎

Submodularity has a feature of the marginal gain diminishing. That is, as the number of agents monitoring a specific node increases, the marginal gain of the joint probability from the node decreases. It is important to appropriately allocate the agents to a given space to satisfy the objective of the problem. In addition, the mathematical problem is

represented as a submodular maximization problem under a partition matroid constraint, which is known to be NP-hard in general.[32,33] This means that the optimal solution can be intractable to obtain within a reasonable time. Solution methodologies to find high-quality solutions should be designed. Therefore, we present an exact solution approach, which is a cutting-plane algorithm with valid inequalities, to solve the problem in Sections 3 and 4. In Section 3, we deal with the mobile multi-agent sensing problem in which the objective value is deterministic. In Section 4, we cover the risk-averse problem by using CVaR.

3 | ALGORITHM FOR THE DETERMINISTIC PROBLEM

In this section, we design a cutting-plane algorithm as an exact solution approach to the mobile multi-agent sensing problem. A cutting-plane algorithm is one of the optimization methods used to efficiently find a (near)-optimal solution by using valid inequalities (optimality cuts). We derive two types of valid inequalities based on the propositions of Nemhauser et al.,[33] which are used to refine the feasible solution set and achieve optimality. Algorithm 1 shows a general procedure of applying a cutting-plane algorithm to an optimization problem ($\max\{f(z) : z \in Z\}$).

Before conducting the algorithm, a relaxed problem ($\max\{g(\sigma) : z \in \mathcal{I}\}$), which is a relaxed version of the original problem, has to be defined. In the first iteration, an optimal solution, $\bar{z}$, and an optimal objective value are obtained after solving the relaxed problem. The optimal objective value is probably infinity because the first iteration proceeds without constraints that are related to σ. Then, valid inequalities consisting of $\bar{z}$ and σ are added to the relaxed problem as constraints. Valid inequalities have to be designed to separate an incumbent solution from the feasible region. Therefore, the generated inequalities lead to reducing a feasible region of σ in the relaxed problem. The algorithm iteratively refines the region until the optimality gap is below ϵ (optimality tolerance).

Algorithm 1. Cutting-plane algorithm

set $\mathcal{I}$ (set of valid inequalities) $\leftarrow \emptyset$, UB (upper bound) $\leftarrow \infty$, LB (lower bound) $\leftarrow -\infty$
while $1 - \frac{LB}{UB} \geq \epsilon$ **do**
  Solve a relaxed problem ($\max\{g(\sigma) : z \in \mathcal{I}\}$);
  Obtain an optimal solution $\bar{z}$;
  $UB \leftarrow$ optimal objective value of the relaxed problem;
  Add valid inequalities (optimality cuts) to $\mathcal{I}$ based on $\bar{z}$;
  if $LB \leq f(\bar{z})$ **then**
    $LB \leftarrow f(\bar{z})$;
  end if
end while

In the mobile multi-agent sensing problem, a feasible set $\overline{S}$ has to satisfy the constraint $|\overline{S} \cap S_i| \leq 1$ for all $i \in X$. We know that agent i has $|S_i|$ strategies. To express a form of integer programming according to the relaxed problem, we introduce a binary decision variable, x_{it}. If agent i selects t^{th} strategy, $x_{it} = 1$; otherwise 0. Using the binary decision variables, $\overline{S}$ is represented as a vector of $\sum_{i \in X} |S_i|$ binary decision variables $(x_{11}, x_{12}, \ldots, x_{1|S_1|}, \ldots, x_{N1}, x_{N2}, \ldots, x_{N|S_N|})$. The constraint $|\overline{S} \cap S_i| \leq 1$ for all $i \in X$ is equal to $\sum_{t=1}^{|S_i|} x_{it} \leq 1$ for all $i \in X$. Because the problem is a maximization problem and the objective function is monotone increasing, we use $\sum_{t=1}^{|S_i|} x_{it} = 1$ for all $i \in X$ instead of $\sum_{t=1}^{|S_i|} x_{it} \leq 1$ for all $i \in X$. For the sake of simplicity, we use $f_j(\overline{S}) := E_j \times P_j(\overline{S})\ \forall j \in Y$ and $f(\overline{S}) := \sum_{j=1}^{N} f_j(\overline{S})$. We define $\rho_k(\overline{S})$ as $f(\overline{S} \cup \{k\}) - f(\overline{S})$. For a given $\overline{S}$, let q_i be the value of t such that $x_{it} = 1$ for agent i. $\overline{S}_i$ denotes the strategy set of agent i ($\overline{S}_i = \{(i, q_i)\}$). $\overline{S} = \overline{S}_1 \cup \overline{S}_2 \cup \cdots \cup \overline{S}_N$ is defined as the strategy set of all agents.

We present a relaxed master problem (MP1) for using a cutting-plane algorithm as follows:

$$[\text{MP1}] \quad \max \sigma \tag{3}$$

$$\text{s.t.} \sum_{t=1}^{|S_i|} x_{it} = 1 \ \forall i \in X \tag{4}$$

$$(\overline{S}, \sigma) \in \mathcal{I} \tag{5}$$

$$x_{it} \in \{0,1\}\ \forall i, t\ \&\ \sigma \in \mathbb{R} \tag{6}$$

where σ is the optimal objective value of MP1, which results in an upper bound of the mobile multi-agent sensing problem for an incumbent solution $\overline{S}$. In the beginning, $\mathcal{I}$ is an empty set and MP1 has a partition matroid constraint only. A feasible solution, $\overline{S}$, is obtained at each iteration, then valid inequalities using $\overline{S}$ are added to MP1 as constraints. Given $\overline{S}$, the objective value of (1), which is equivalent to $f(\overline{S})$, becomes a lower bound. Constraint (4) is equivalent to a partition matroid constraint. In Constraint (5), $\mathcal{I}$ means a set of valid inequalities. The generated inequalities lead to reducing the feasible region in MP1.

For a given set $\overline{S}$, we propose two valid inequalities (optimality cuts) related to σ as follows:

$$\sigma \le f(\overline{S}) + \sum_{i=1}^{N} \rho_{a_i}(\overline{S}) \times (1 - x_{iq_i})$$
$$(a_i = \underset{a}{\operatorname{argmax}}\, \rho_a(\overline{S}) \text{ such that } a \in S_i\ \forall i \in X) \tag{7}$$

$$\sigma \le f(\overline{S}) + \sum_{i=1}^{N} \rho_{b_i}(\emptyset) \times (1 - x_{iq_i}) - \sum_{i=1}^{N} \rho_{\overline{S}_i}(\overline{S} \setminus \overline{S}_i) \times (1 - x_{iq_i})$$
$$(b_i = \underset{b}{\operatorname{argmax}}\, \rho_b(\emptyset) \text{ such that } b \in S_i\ \forall i \in X) \tag{8}$$

Theorem 2. *For a given feasible set $\overline{S}$, Inequality (7) is valid for MP1.*

Proof. Consider two feasible sets $\overline{S}^A$ and $\overline{S}^B$. Proposition 2.1.(v) of Nemhauser et al. shows that f is submodular if and only if $f(\overline{S}^B) \le f(\overline{S}^A) + \sum_{a \in \overline{S}^B \setminus \overline{S}^A} \rho_a(\overline{S}^A)$.[33] We show that Inequality (7) is valid for MP1 by using Proposition 2.1.(v) of Nemhauser et al.[33]

We assume that there is an arbitrary feasible solution $(\hat{S}, \hat{\sigma})$ for MP1. We know that $\hat{\sigma} \le f(\hat{S})$ satisfies a feasibility condition for MP1.

$$\begin{aligned}
\hat{\sigma} &\le f(\hat{S}) \\
&\le f(\overline{S}) + \sum_{a \in \hat{S} \setminus \overline{S}} \rho_a(\overline{S}) \qquad (9)\\
&= f(\overline{S}) + \sum_{i=1}^{N} \sum_{a \in \hat{S}_i \setminus \overline{S}_i} \rho_a(\overline{S}) \\
&= f(\overline{S}) + \sum_{i=1}^{N} \sum_{a \in \hat{S}_i \setminus \overline{S}_i} \rho_a(\overline{S}) \times \left(\sum_{t=1}^{|S_i|} x_{it} - x_{iq_i} \right) \\
&\le f(\overline{S}) + \sum_{i=1}^{N} \rho_{a_i}(\overline{S}) \times \left(\sum_{t=1}^{|S_i|} x_{it} - x_{iq_i} \right) \\
&= f(\overline{S}) + \sum_{i=1}^{N} \rho_{a_i}(\overline{S}) \times (1 - x_{iq_i}). \qquad (10)
\end{aligned}$$

Inequality (9) is due to Proposition 2.1. (v) of Nemhauser et al.[33] The first equality in Inequality (9) follows from $\hat{S} \setminus \overline{S} = \bigcup_{i=1}^{N} (\hat{S}_i \setminus \overline{S}_i)$. In the second equality, if $\hat{S}_i \ne \overline{S}_i$ for some i, x_{iq_i} becomes 0. We know $\sum_{t=1}^{|S_i|} x_{it} = 1$, therefore, the equality is satisfied. Inequality (10) follows from the definition of a_i. Therefore, Inequality (7) becomes an optimality cut for MP1 when a feasible set $\overline{S}$ is given. ∎

Theorem 3. *For a given feasible set $\overline{S}$, Inequality (8) is valid for MP1.*

Proof. Consider two feasible sets $\overline{S}^A$ and $\overline{S}^B$. Proposition 2.1(vii) of Nemhauser et al. shows that f is submodular if and only if $f(\overline{S}^B) \le f(\overline{S}^A) + \sum_{b \in \overline{S}^B \setminus \overline{S}^A} \rho_b(\overline{S}^A \cap \overline{S}^B) - \sum_{b \in \overline{S}^A \setminus \overline{S}^B} \rho_b(\overline{S}^A \setminus \{b\})$.[33] We show that Inequality (8) is valid for MP1 by using Proposition 2.1.(vii) of Nemhauser et al.[33]

We assume that there is an arbitrary feasible solution $(\hat{S}, \hat{\sigma})$ for MP1. We know that $\hat{\sigma} \le f(\hat{S})$ satisfies a feasibility condition for MP1.

$$\begin{aligned}
\hat{\sigma} &\le f(\hat{S}) \\
&\le f(\overline{S}) + \sum_{b \in \hat{S} \setminus \overline{S}} \rho_b(\overline{S} \cap \hat{S}) - \sum_{b \in \overline{S} \setminus \hat{S}} \rho_b(\overline{S} \setminus \{b\}) && (11) \\
&\le f(\overline{S}) + \sum_{b \in \hat{S} \setminus \overline{S}} \rho_b(\emptyset) - \sum_{b \in \overline{S} \setminus \hat{S}} \rho_b(\overline{S} \setminus \{b\}) && (12) \\
&= f(\overline{S}) + \sum_{i=1}^{N} \sum_{b \in \hat{S}_i \setminus \overline{S}_i} \rho_b(\emptyset) - \left(\sum_{i=1}^{N} \sum_{b \in \overline{S}_i} \rho_b(\overline{S} \setminus \{b\}) - \sum_{i=1}^{N} \sum_{b \in \overline{S}_i \cap \hat{S}_i} \rho_b(\overline{S} \setminus \{b\}) \right) \\
&= f(\overline{S}) + \sum_{i=1}^{N} \sum_{b \in \hat{S}_i \setminus \overline{S}_i} \rho_b(\emptyset) \times \left(\sum_{t=1}^{|S_i|} x_{it} - x_{iq_i} \right) \\
&\quad - \left(\sum_{i=1}^{N} \sum_{b \in \overline{S}_i} \rho_b(\overline{S} \setminus \{b\}) - \sum_{i=1}^{N} \sum_{b \in \overline{S}_i \cap \hat{S}_i} \rho_b(\overline{S} \setminus \{b\}) \times x_{iq_i} \right) \\
&\le f(\overline{S}) + \sum_{i=1}^{N} \rho_{b_i}(\emptyset) \times (1 - x_{iq_i}) \\
&\quad - \left(\sum_{i=1}^{N} \sum_{b \in \overline{S}_i} \rho_b(\overline{S} \setminus \{b\}) - \sum_{i=1}^{N} \sum_{b \in \overline{S}_i \cap \hat{S}_i} \rho_b(\overline{S} \setminus \{b\}) \times x_{iq_i} \right) && (13) \\
&\le f(\overline{S}) + \sum_{i=1}^{N} \rho_{b_i}(\emptyset) \times (1 - x_{iq_i}) \\
&\quad - \left(\sum_{i=1}^{N} \sum_{b \in \overline{S}_i} \rho_b(\overline{S} \setminus \{b\}) - \sum_{i=1}^{N} \sum_{b \in \overline{S}_i} \rho_b(\overline{S} \setminus \{b\}) \times x_{iq_i} \right) \\
&= f(\overline{S}) + \sum_{i=1}^{N} \rho_{b_i}(\emptyset) \times (1 - x_{iq_i}) - \sum_{i=1}^{N} \rho_{\overline{S}_i}(\overline{S} \setminus \overline{S}_i) \times (1 - x_{iq_i}). && (14)
\end{aligned}$$

Inequality (11) is due to Proposition 2.1.(vii) of Nemhauser et al.[33] Inequality (12) follows from the submodularity of f. The first equality in Inequality (12) follows from $\hat{S} \setminus \overline{S} = \bigcup_{i=1}^{N} (\hat{S}_i \setminus \overline{S}_i)$ and $\hat{S}_i \setminus \overline{S}_i = \hat{S}_i - (\hat{S}_i \cap \overline{S}_i)$. In the second equality, if $\hat{S}_i \neq \overline{S}_i$ for some i, x_{iq_i} becomes 0. We know $\sum_{t=1}^{|S_i|} x_{it} = 1$, therefore, the equality is satisfied. Inequality (13) follows from the definition of b_i. Inequality (14) is due to $\overline{S}_i \cap \hat{S}_i \subseteq \overline{S}_i$. Therefore, Inequality (8) becomes an optimality cut for MP1 when a feasible set $\overline{S}$ is given. ∎

Algorithm 2 shows a procedure of the cutting-plane algorithm with the two valid inequalities (Inequalities (7) and (8)) to solve the mobile multi-agent sensing problem.

In each iteration, we solve MP1 to obtain an incumbent solution $(\overline{S}, \overline{\sigma})$. The optimal objective value of MP1 $(\overline{\sigma})$ becomes the upper bound for the problem. Using the strategy set $\overline{S}$, we add two valid inequality to MP1. If $LB \le f(\overline{S})$, the lower bound becomes $f(\overline{S})$. If the gap between the upper and lower bounds is smaller than ϵ, the algorithm is terminated.

The computation time of Algorithm 2 is affected depending on which initial solution is used before solving MP1. Depending on an initial solution, we may start the algorithm with a high lower bound, and more effective valid inequalities may be obtained. In Section 5, we analyze the efficiency of Algorithm 2 depending on different types of initial solutions.

Algorithm 2 is designed to solve the mobile multi-agent sensing problem when the objective value $f(\overline{S})$ is deterministic. In reality, agents might break down or not detect the target accurately. This means that there might be uncertainty in the mobile multi-agent sensing problem. In Section 4, we consider the risk-averse problem by using CVaR in the objective function.

Algorithm 2. Cutting-plane algorithm for the deterministic problem

Input: $S(= S_1 \cup S_2 \cup \cdots \cup S_N$, S_i: set of all strategies for agent $i)$
Output: $\overline{S}$
set $\mathcal{I} \leftarrow \emptyset$, $UB \leftarrow \infty$, $LB \leftarrow -\infty$
while $1 - \frac{LB}{UB} \geq \epsilon$ **do**
  Solve MP1;
  $(\overline{S}, \overline{\sigma}) \leftarrow$ optimal solution for MP1;
  $UB \leftarrow \overline{\sigma}$;
  Add two valid inequalities ((7) and (8)) to $\mathcal{I}$ based on $\overline{S}$;
  if $LB \leq f(\overline{S})$ **then**
    $LB \leftarrow f(\overline{S})$;
  end if
end while

4 | ALGORITHM FOR THE RISK-AVERSE PROBLEM

In this section, we present the mobile multi-agent sensing problem with uncertain situations. We defined $f(\overline{S}) := \sum_{j=1}^{N} f_j(\overline{S})$ $(f_j(\overline{S}) := E_j \times P_j(\overline{S})$ $\forall j \in Y)$ as an objective function of the deterministic problem in Section 3. To reflect uncertain situations, we introduced a new objective function, $f(\overline{S}, \omega)$. $\omega \in \Omega$ is a random variable that represents uncertainty and is also independent of $\overline{S}$. We consider that $f(\overline{S}, \omega)$ is monotone and submodular. A basic approach to the problem with uncertainty is to use the expected value of the objective function $\mathbb{E}_{\omega}[f(\overline{S}, \omega)]$. However, in the case of high variability, there is a high possibility that the basic approach may find poor solutions. Instead of this approach, recent research has focused on a risk-averse submodular optimization to reflect the risk from the uncertainty in a rational way.[46,47]

The risk-averse decisions are important when failure from uncertainty leads to severe or catastrophic consequences. In a finance field, *value-at-risk* (VaR) is known as a popular risk measure:

$$VaR_{\alpha}\big(f(\overline{S}, \omega)\big) = \inf\{\tau : \text{Prob}\big[f(\overline{S}, \omega) \leq \tau\big] \geq \alpha, \tau \in \mathbb{R}\}. \tag{15}$$

For a given risk level $\alpha \in (0, 1]$, VaR means the (left) α-percentile of the random variable $f(\overline{S}, \omega)$. However, *conditional value-at-risk* (CVaR) is the expectation of $f(\overline{S}, \omega)$ from α-percentile cases of $f(\overline{S}, \omega)$. CVaR is a more popular measure than VaR because CVaR has some properties (coherence and tractability).[49] Because CVaR is more conservative than VaR, more risk-averse solutions are obtained.[45]

In this article, we use CVaR as a risk measure. For a given risk level $\alpha \in (0, 1]$, the definition of CVaR is as follows:

$$CVaR_{\alpha}\big(f(\overline{S}, \omega)\big) = \mathbb{E}\big[f(\overline{S}, \omega) | f(\overline{S}, \omega) \leq VaR_{\alpha}(f(\overline{S}, \omega))\big]. \tag{16}$$

CVaR is not necessarily a submodular function. We use an auxiliary function to maintain submodularity. It is known that maximizing $CVaR_{\alpha}(\tau, f(\overline{S}, \omega))$ over $\overline{S}$ is equivalent to maximizing $\tau - \frac{1}{\alpha}\mathbb{E}\big(\big[\tau - f(\overline{S}, \omega)\big]_{+}\big)$ over $\overline{S}$ and τ:[45,47,50]

$$\max_{\overline{S}} CVaR_{\alpha}\big(\tau, f(\overline{S}, \omega)\big) = \max_{\overline{S}, \tau} \tau - \frac{1}{\alpha}\mathbb{E}\big(\big[\tau - f(\overline{S}, \omega)\big]_{+}\big), \tag{17}$$

where $[\tau - f(\overline{S}, \omega)]_{+}$ is defined as $\max(\tau - f(\overline{S}, \omega), 0)$. The auxiliary function is used to solve the problem. The mobile multi-agent sensing problem that maximizes CVaR at a given risk level $\alpha \in (0, 1]$ is expressed as follows:

$$\max_{\overline{S}, \tau} \ \tau - \frac{1}{\alpha}\mathbb{E}\left([\tau - f(\overline{S}, \omega)]_{+}\right), \tag{18}$$

$$\text{subject to } |\overline{S} \cap S_i| \leq 1, \forall i \in X \text{ and } \tau \in \mathbb{R}. \tag{19}$$

As the number of agents increases or the sensing environment gets more complex, the number of scenarios that result from the uncertainty might increase exponentially. It may take a great deal of time to solve the problem by using the proposed exact algorithm for a large data set or complex environment. The sample average approximation approach is introduced to approximately estimate the objective value of stochastic optimization problems by using finite scenarios derived from a random sample. In this article, the problem of maximizing CVaR based on the sample average approximation is considered.

We assume that there is a set of finite scenarios $\Omega = \{\omega_1, \omega_2, \dots, \omega_K\} \subset \mathcal{W}$ and the probability of each scenario ω_k is p_k. We define $\rho_a^k(\overline{S})$ as $f(\overline{S} \cup \{a\}, \omega_k) - f(\overline{S}, \omega_k)$. Based on the sample average approximation approach, we present a relaxed master problem (MP2) that maximizes CVaR at a given risk level, $\alpha \in (0, 1]$, as follows:

$$[\text{MP2}] \ \max \ \tau - \frac{1}{\alpha} \sum_{k \in [K]} p_k r_k, \tag{20}$$

$$\text{s.t.} \sum_{t=1}^{|S_i|} x_{it} = 1 \ \forall i \in X, \tag{21}$$

$$r_k \geq \tau - \sigma_k \ \forall k \in [K], \tag{22}$$

$$(\overline{S}, \sigma_1, \sigma_2, \dots, \sigma_K) \in \mathcal{I}, \tag{23}$$

$$x_{it} \in \{0, 1\} \ \forall i, t, \ r \in \mathbb{R}_+^K \ \tau \in \mathbb{R} \ \sigma \in \mathbb{R}^K, \tag{24}$$

where $[K] = \{1, 2, \dots, K\}$. The objective value of MP2 is an upper bound of the problem that maximizes CVaR for an incumbent solution $\overline{S}$. r_k and σ_k are represented as $(\tau - f(\overline{S}, \omega_k))_+$ and $f(\overline{S}, \omega_k)$, respectively. Given $\overline{S}$ and τ, we can calculate the objective value of Problem (18). The objective value becomes a lower bound because $\overline{S}$ and τ is a feasible solution of Problem (18). Constraint (21) is equivalent to a partition matroid constraint. In Constraint (23), $\mathcal{I}$ means a set of valid inequalities. In the beginning, $\mathcal{I}$ is an empty set. A feasible solution, $\overline{S}$, is obtained at each iteration, then valid inequalities using $\overline{S}$ are added to MP2 as constraints. The generated inequalities lead to reducing the feasible region in MP2.

For a given set $\overline{S}$ and scenario ω_k, we propose two valid inequalities (optimality cuts) related to $\sigma_1, \sigma_2, \dots, \sigma_K$ as follows:

$$\sigma_k \leq f\left(\overline{S}, \omega_k\right) + \sum_{i=1}^{N} \rho_{a_i^k}^k\left(\overline{S}\right) \times (1 - x_{iq_i})$$

$$(a_i^k = \underset{a}{\text{argmax}} \, \rho_a^k(\overline{S}) \text{ such that } a \in S_i \ \forall i \in X), \tag{25}$$

$$\sigma_k \leq f(\overline{S}, \omega_k) + \sum_{i=1}^{N} \rho_{b_i^k}^k(\emptyset) \times (1 - x_{iq_i}) - \sum_{i=1}^{N} \rho_{S_i}^k\left(\overline{S} \setminus \overline{S}_i\right) \times (1 - x_{iq_i})$$

$$(b_i^k = \underset{b}{\text{argmax}} \, \rho_b^k(\emptyset) \text{ such that } b \in S_i \ \forall i \in X), \tag{26}$$

The above two inequalities are based on Inequalities (7) and (8). Algorithm 3 shows a procedure of the cutting-plane algorithm using the two valid inequalities (Inequalities (25) and (26)) to solve the problem of maximizing CVaR based on the sample average approximation.

In each iteration, we solve MP2 to obtain an incumbent solution $(\overline{S}, \overline{\tau}, \overline{r}, \overline{\sigma})$. The optimal objective value of MP2 becomes the upper bound. Using the strategy set $\overline{S}$, we add two valid inequalities to MP2 for each scenario. This means that $2K$ constraints are generated at each iteration. After generating cuts, the algorithm solves the problem (18) in which the strategy set $\overline{S}$ is given. If the objective value of the problem (18) is higher than LB, the lower bound is updated as the objective value of the problem (18). If the gap between the upper and lower bounds is smaller than ϵ, the algorithm is terminated and we obtain the (near)-optimal strategy.

Algorithm 3. Cutting-plane algorithm for the risk-averse problem

Input: $\mathcal{S}(=\mathcal{S}_1 \cup \mathcal{S}_2 \cup \cdots \cup \mathcal{S}_N$, $\mathcal{S}_i$: set of all strategies for agent i), $\Omega = \{\omega_1, \omega_2, \ldots, \omega_K\}$: set of finite scenarios
Output: $\overline{\mathcal{S}}$
set $\mathcal{I} \leftarrow \emptyset$, $UB \leftarrow \infty$, $LB \leftarrow -\infty$
while $1 - \frac{LB}{UB} \geq \epsilon$ **do**
    Solve MP2;
    $(\overline{\mathcal{S}}, \overline{\tau}, \overline{r}, \overline{\sigma}) \leftarrow$ optimal solution for MP2;
    $UB \leftarrow$ objective value for MP2;
    $k \leftarrow 1$;
    while $k \leq K$ **do**
        Add two valid inequalities ((25) and (26)) to $\mathcal{I}$ based on $\overline{\mathcal{S}}$ & ω_k;
        $k \leftarrow k + 1$;
    end while
    Solve the problem (18) in which the strategy set $\overline{\mathcal{S}}$ is given;
    $LB_{cur} \leftarrow$ objective value for the problem (18);
    if $LB \leq LB_{cur}$ **then**
        $LB \leftarrow LB_{cur}$;
    end if
end while

TABLE 2 Parameter sets

Parameter	Value
N	10, 20, 30, 40, 50
M	10, 20, 30, 40, 50, 60, 80, 100
cl_i	$[0,100]^2$
o_j	$[0,100]^2$
L_i	(Integer) Uniform[1, 5]
E_j	Uniform(0, 1)
δ_i	(Integer) 5×Uniform[1, 4]
λ_i	0.4
ϵ	0.1
α	0.1 / 0.01 & 1.0 (Figure 3)

5 | COMPUTATIONAL EXPERIMENTS

In this section, computational experiments are conducted to show the performance of Algorithms 2 and 3. First of all, in Subsection 5.1, we conduct numerical experiments with Algorithm 2 for the deterministic problem. In Subsection 5.2, we present numerical results of Algorithm 3 for the problem of maximizing CVaR based on the sample average approximation. All tests were run on a Python 3 and a CPLEX ver.12.10 with an Intel core CPU i5-3470 processor. We considered a realistic situation in which there are a large number of M nodes and a relatively small number of N agents. The agents and nodes are located as points, which are generated as uniformly random, in a two-dimensional space, $\mathbb{R}^2$. The algorithms are terminated when the ratio of $\frac{LB}{UB}$ achieves $(1-\epsilon)\times 100\%$. The time limit was set to 3600 s. In real cases, it is possible for decision makers to adjust the time limit based on the decision period and the number of agents and nodes. We defined a strategy set, $\mathcal{S}_i$, of agent i as all points whose distances to cl_i are within L_i. We restricted the points to integers; otherwise, the number of strategy sets is infinite. Parameter values used in the numerical experiments are presented in Table 2.

TABLE 3 Results for the two guarantees and the optimality gap

		Guarantees or optimality gap		
N	***M***	**Qu et al.[39]**	**Lee et al.[38]**	**Algorithm 2 (1st iter.)**
10	20	0.505	0.641	0.939
20	40	0.508	0.610	0.912
30	60	0.510	0.617	0.866
40	80	0.519	0.603	0.809
50	100	0.515	0.605	0.786

When conducting the cutting-plane algorithm, it is important to decide the way to get an initial solution so that the algorithm can find (near)-optimal solutions efficiently. We considered three methods of choosing initial solutions in these experiments. The three methods are as follows:

(a) *Random*: a randomly generated solution that satisfies the constraints
(b) *Individual*: a solution in which each agent chooses its optimal strategy without considering others' strategies
(c) *Sequential*: a solution in which a greedy algorithm is applied according to a fixed order of agents

In the *Sequential* method, we first set an order of agents, and then select a strategy of each agent in which marginal gain gets maximized from the current situation according to the order (which corresponds to a greedy algorithm). The Sequential method, which is a variant of a greedy algorithm, is known to have approximation ratios for the proposed problem. In particular, Qu et al. and Lee et al. introduced theoretical instance-dependent guarantees in terms of approximation ratios.[38,39] The guarantees (ratios) are dependent on instances. The guarantee means the relative difference between the upper bound and value obtained from the Sequential method. Thus, it is interpreted as an optimality gap for the problem.

A different optimality gap of the same solution can also be obtained through the first iteration in Algorithm 2. Before presenting the performances of the algorithms, we compared the two guarantees and the optimality gap of the first iteration in Algorithm 2 in terms of the effectiveness of the upper bounds. We executed 10 runs for each M and N, and Table 3 shows the results for the two guarantees and the optimality gap.

The guarantees introduced by Qu et al. showed up as being slightly larger than 0.5,[39] and the guarantees introduced by Lee et al. showed up as being close to 0.6,[38] on average. On the other hand, the optimality gap in Algorithm 2 was larger than the two guarantees by 18% to 43% in all cases. Even though the same solutions from the *Sequential* method were obtained, the algorithm in this article showed more effective upper bounds than the two guarantees.

5.1 | Experiments for the deterministic problem

First of all, we conducted numerical experiments to analyze the performance of Algorithm 2 for the deterministic problem. Table 4 shows the results for different values of N and M (five cases). We executed 10 runs for each case, and the values in Table 4 are all average values. Column *vars* denotes the range of the number of decision variables used in MP1. Column *iters* means the number of iterations before the algorithm was terminated. Column *cuts* denotes the number of valid inequalities (optimality cuts) added to MP1. Column *time* means the computation times (in) that Algorithm 2 took. Numbers in parentheses of column *time* denote the number of instances that Algorithm 2 solved within the time limit.

For small data sets ($(N, M) = (10, 20)$ and $(20, 40)$), all instances were solved within the time limit. When we applied *Individual* and *Sequential* for initial solutions, the number of iterations was one for all instances. On the other hand, in *Random*, the algorithm conducted more than 13 iterations to find (near)-optimal solutions. In complex situations, we observed that the computation time increased as the number of N and M increased for all three methods. Moreover, the number of instances solved within 3,600 tended to reduce as the number of N and M increased. In the instances not solved within 3600 s, the average ratio of $\frac{LB}{UB}$ showed 79%, 85%, and 86% at *Random*, *Individual*, and *Sequential*. For all instances, we observed that the performances in *Individual* and *Sequential* were better compared to *Random*. Therefore, choosing initial solutions properly is crucial so that the algorithm can find (near)-optimal solutions efficiently.

TABLE 4 Results for the different values N and M

			Random			Individual			Sequential		
N	M	Vars	Iters	Cuts	Time	Iters	Cuts	Time	Iters	Cuts	Time
10	20	110~210	13.0	26.0	4 (10)	1.0	2.0	1 (10)	1.0	2.0	1 (10)
20	40	220~420	17.0	34.0	33 (10)	1.0	2.0	5 (10)	1.0	2.0	6 (10)
30	60	330~630	36.5	73.0	212 (8)	3.0	6.0	24 (8)	3.0	6.0	27 (8)
40	80	440~840	144.0	288.0	1756 (4)	12.0	24.0	156 (8)	9.5	19.0	132 (8)
50	100	550~1050	86.0	172.0	2306 (1)	2.0	4.0	68 (1)	47.5	95.0	1147 (2)

TABLE 5 Results for the different values M

			Random			Individual			Sequential		
N	M	Vars	Iters	Cuts	Time	Iters	Cuts	Time	Iters	Cuts	Time
20	10	220~420	15.0	30.0	9	3.0	6.0	2	3.0	6.0	3
20	20	220~420	34.5	69.0	34	5.5	11.0	7	5.5	11.0	7
20	30	220~420	57.0	114.0	81	17.0	34.0	24	12.5	25.0	19
20	40	220~420	54.5	109.0	102	10.5	21.0	21	7.5	15.0	17
20	50	220~420	152.0	304.0	425	70.5	141.0	167	27.0	54.0	67

TABLE 6 Results for the different values N

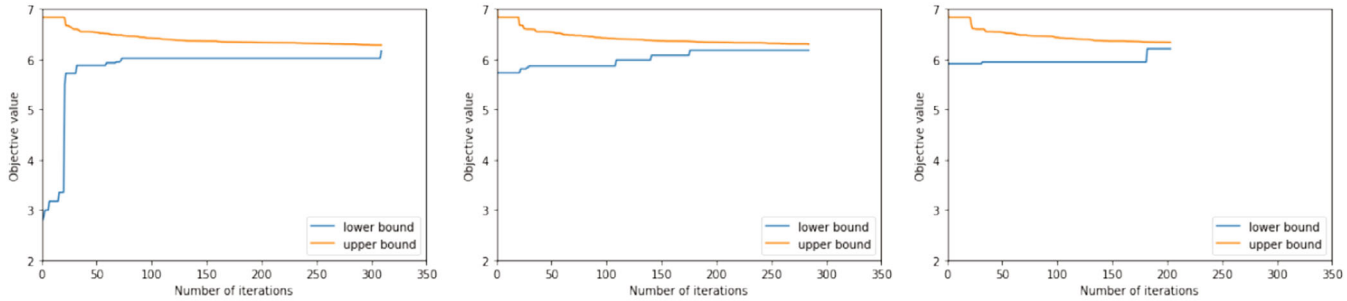
			Individual			Sequential		
N	M	Vars	Iters	Cuts	Time	Iters	Cuts	Time
10	50	110~210	45.6	91.2	19 (10)	43.0	86.0	19 (10)
20	50	220~420	17.1	34.2	26 (10)	9.7	19.4	16 (10)
30	50	330~630	24.8	49.6	58 (10)	37.8	75.6	85 (10)
40	50	440~840	52.2	104.4	760 (5)	91.7	183.4	1340 (4)
50	50	550~1050	53.0	106.0	1156 (1)	27.5	55.0	620 (2)

To consider the situation in which the number of agents is fixed, we conducted experiments (five cases) depending on the number of nodes with a fixed number of agents $N = 20$. The number of agents that managers have may be limited because of purchasing costs, operating costs, and other factors. The results are summarized in Table 5.

The range of the total number of the decision variables in Table 2 was identical because it is determined depending on the number of agents. Even though the range was identical, as the number of nodes increased, both the number of cuts added and the computation times attained tended to increase (except for (20, 40) instances). As the number of nodes increased, it was more likely that the coefficients of the two valid inequalities $\left(\sum_{i=1}^{N}\rho_{a_i}(\overline{S})\right.$ and $\sum_{i=1}^{N}\rho_{b_i}(\emptyset) - \sum_{i=1}^{N}\rho_{\overline{S}_i}(\overline{S} \setminus \left.\overline{S}_i)\right)$ got large. In fact, the upper bound given by the algorithm had to decrease, according to the added valid inequalities. If the coefficients were large, the decreasing rate of the upper bound might be small at each iteration.

In all cases, we observed more efficient results when we applied *Individual* and *Sequential* compared to *Random*. When the number of nodes was small, there was no significant difference in the performance between *Individual* and *Sequential*. However, as the number of nodes increased, *Sequential* could find (near)-optimal solutions with a relatively small number of cuts, compared to *Individual*.

The *Random* method for determining an initial solution was poor in terms of computation times and ratios, so we covered only the other two methods for the next experiments. We handled a specific situation in which the number of nodes was fixed to 50. Given a fixed number of nodes in a specific area, managers might determine the number of agents to use by considering operating costs, legal issues, and other factors. The results are summarized in Table 6.



(A) *Random* method (B) *Individual* method (C) *Sequential* method

FIGURE 1 Result for Algorithm 2

The values in Table 6 are all average values. We executed 10 runs for each N and M. As the number of agents increased, the number of decision variables increased as well. This means that the number of strategies considered gets large. As a result, the computation time to obtain the (near)-optimal solution with a ratio of $(1-\epsilon)\times 100\%$ tended to increase. Also, in a complex situation, the algorithm might need a lot of optimality cuts to lower its current upper bound. As the number of agents increased, the instances of solving the problems within the time limit of 3600 s decreased. In the instances that were not solved within 3600 s, the average ratio of $\frac{LB}{UB}$ showed 85% at both *Individual* and *Sequential*.

Figures 1A–1C show the changing shape of the upper and lower bounds according to the number of optimality cuts added. The figures represent the results for Algorithm 2 with the *Random*, *Individual*, and *Sequential* methods, respectively. The three results are analyzed by using the same data. We used an instance for $N = 10$, $M = 50$, and 210 decision variables. The algorithm was terminated when the ratio of $\frac{LB}{UB}$ was less than 2%. In all figures, as the number of cuts increased, the lower bound gradually increased while the upper bound gradually decreased. In particular, the upper bound shrunk little by little in every iteration, while the lower bound increased occasionally but significantly.

Figure 1a shows the case of the *Random* method. The algorithm found a (near)-optimal solution after 628 cuts were generated. Figure 1b shows the case of the *Individual* method. The algorithm found a (near)-optimal solution after 580 cuts were generated. Figure 1c shows the case of the *Sequential* method. The algorithm found a (near)-optimal solution after 418 cuts were generated. In this instance, as the initial solution given by the *Sequential* method was closer to the optimal solution than it was in other methods, the algorithm might find a (near)-optimal solution with a relatively small number of cuts.

As the number of cuts increased gradually, the upper and lower bounds eventually converged to the optimal solution.

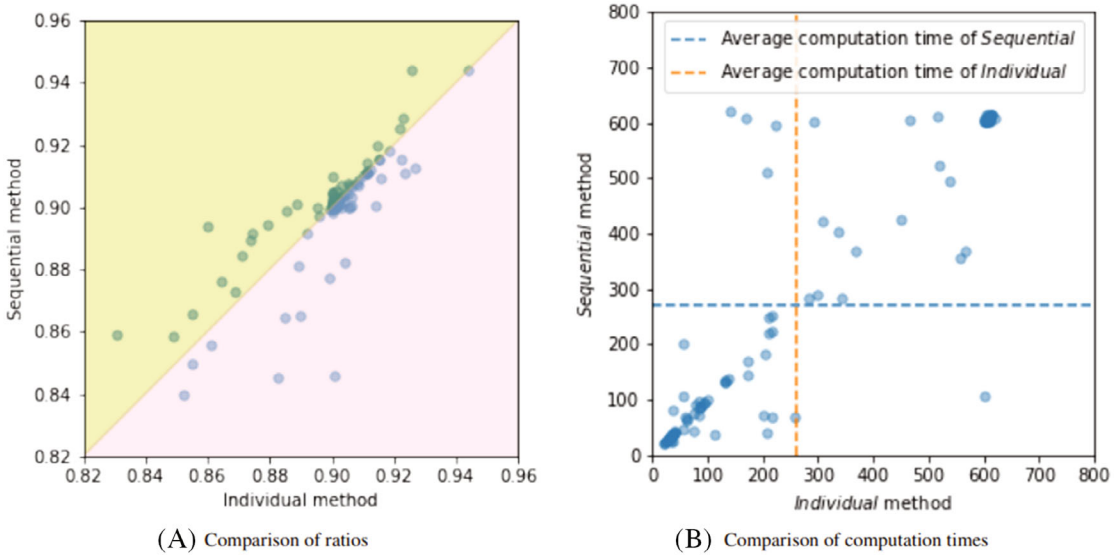
5.2 | Experiments for the risk-averse problem

We conducted numerical experiments on the risk-averse problem discussed in Section 4. We applied Algorithm 3 to solve the risk-averse problems and analyzed the performance of Algorithm 3 for the risk-averse problem. In these experiments, we set $\alpha = 0.1$. We assumed that the failure rate for each agent is 10%, which means that each agent does not work at a 10% probability. We generated $|\Omega| = 15$ scenarios for the sample average approximation approach. We executed 10 runs for each N and M (five cases). Table 7 shows the results for different values of N and M. The values in Table 7 are all average values. Column *vars* denotes the range of the number of the decision variables used in MP2. Column *iters* means the number of iterations before the algorithm was terminated. Column *cuts* denotes the number of valid inequalities (optimality cuts) added to MP2. Column *time* means the computation times (in seconds) that Algorithm 3 took. Numbers in parentheses of column *time* denote the number of instances that Algorithm 3 solved within the time limit.

Because we generate the two valid inequalities (Inequalities (25) and (26)) for each scenario, the algorithm adds 30 ($=2\times 15$) cuts per each iteration during the procedure. The initial solutions given by *Individual* and *Sequential* had no significant difference in the performance. When $N = 50$ and $M = 100$, the instances in which the (near)-optimal solutions were obtained within 3600 s was 3 out of 10 instances in both methods. In the instances not solved within 3600 s, the average ratios of $\frac{LB}{UB}$ showed 84% at both *Individual* and *Sequential*. Compared to the results for Algorithm 2 to solve the deterministic problem (Table 4), the number of iterations of Algorithm 3 was smaller, but the computation time of

TABLE 7 Results for the different values N and M

			Individual			Sequential		
N	M	Vars	Iters	Cuts	Time	Iters	Cuts	Time
10	20	110~210	1.1	33.0	12 (10)	1.1	33.0	12 (10)
20	40	220~420	7.7	231.0	205 (10)	7.9	237.0	213 (10)
30	60	330~630	10.4	312.0	771 (10)	8.6	258.0	641 (10)
40	80	440~840	6.4	193.3	960 (9)	6.3	190.0	952 (9)
50	100	550~1050	5.0	150.0	1481 (3)	5.0	150.0	1481 (3)



(A) Comparison of ratios (B) Comparison of computation times

FIGURE 2 Comparison between *Individual* and *Sequential* methods

TABLE 8 Results for the different values M

			Individual			Sequential		
N	M	Vars	Iters	Cuts	Time	Iters	Cuts	Time
20	10	220~420	2.0	60.0	17	2.0	60.0	17
20	20	220~420	2.0	60.0	36	2.6	78.0	44
20	30	220~420	1.0	30.0	33	1.0	30.0	33
20	40	220~420	7.7	231.0	205	7.9	237.0	213
20	50	220~420	1.2	36.0	65	2.0	60.0	92

Algorithm 3 was larger. This implies that Algorithm 3 took a lot of time to solve MP2 at every iteration because a relatively large number of constraints were added.

Figure 2A is a plot of the ratios of $\frac{LB}{UB}$ obtained by *Individual* and *Sequential* methods when $N = 20$ and $M = 50$. We used 100 instances for the two methods. The dots (instances) in the yellow region indicate that better performances are shown in the *Sequential* method. The dots (instances) in the red region indicate that better performances are shown in the *Individual* method. That is, the *Sequential* method obtained a higher ratio in 60 experiments. Figure 2B is a plot of the computation times obtained by *Individual* and *Sequential* methods. The average computation times of *Individual* and *Sequential* methods were 262 and 270 s, respectively. There was no significant difference in the computation time of the two methods. Therefore, in these experiments, we observed that the use of the *Sequential* method to determine an initial solution was more advantageous in terms of the computation time and the ratio.

TABLE 9 Results for the different values N

			Individual			Sequential		
N	M	Vars	Iters	Cuts	Time	Iters	Cuts	Time
10	50	110~210	1.1	33.0	32 (10)	1.1	33.0	32 (10)
20	50	220~420	1.2	36.0	65 (10)	2.0	60.0	92 (10)
30	50	330~630	11.2	336.0	702 (10)	11.4	342.0	700 (10)
40	50	440~840	4.9	146.3	960 (8)	4.6	138.8	952 (8)
50	50	550~1050	3.1	94.3	1481 (7)	3.1	94.3	1481 (6)

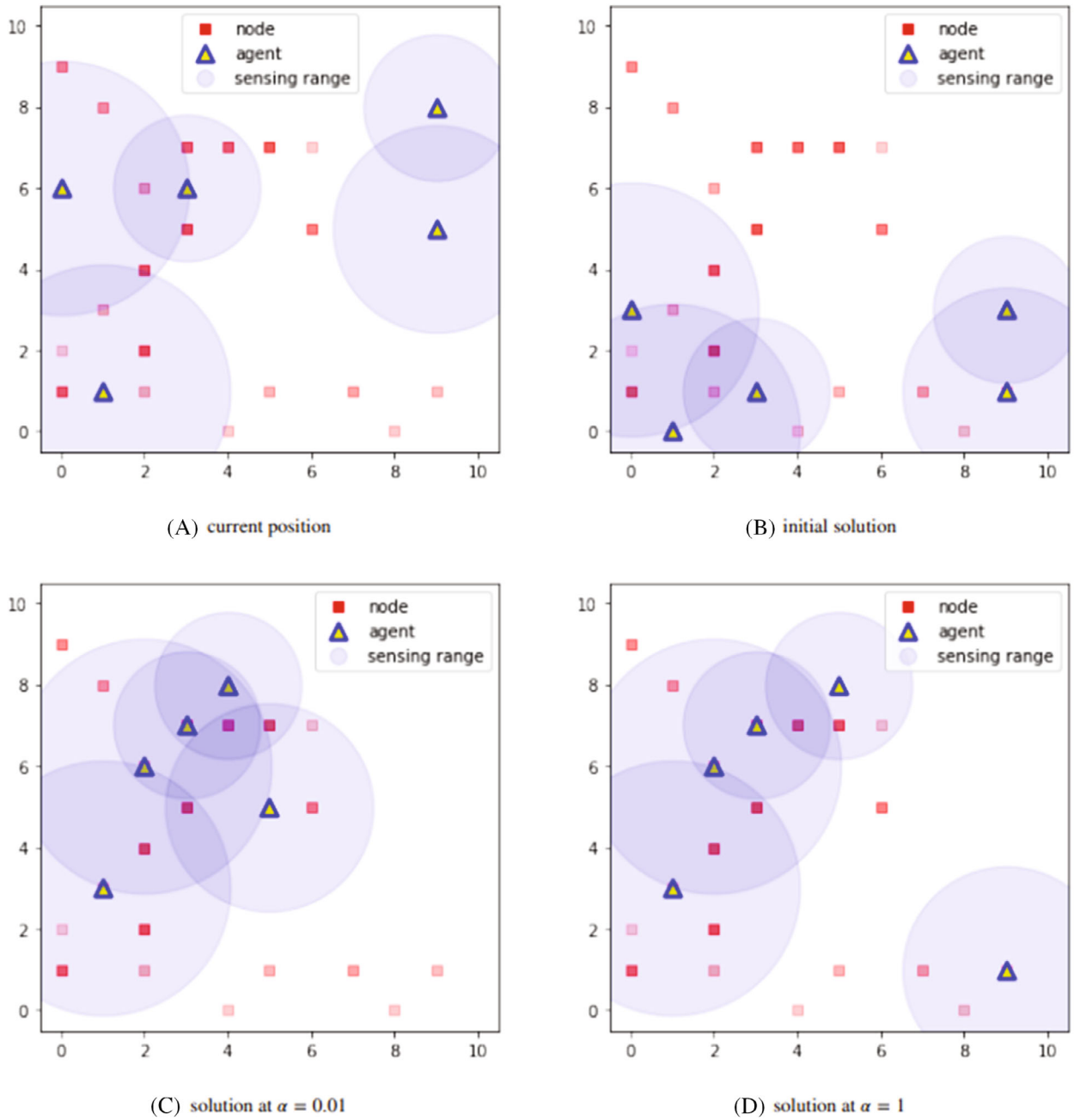


FIGURE 3 Result for Algorithm 3 in terms of a risk level α

We set a specific case in which the number of nodes was fixed to 50 and the number of agents was changed (five cases). The results are summarized in Table 8. The values in Table 8 are all average values from the numerical experiments when we executed 10 runs for each case. Despite maintaining the same experimental environment in Table 5, the results were somewhat different. Compared to Table 5, an increase in the number of iterations or valid inequalities was not significant as the number of nodes increased. In fact, as the number of nodes increased, it was more likely that the coefficients of the two valid inequalities $\left(\sum_{i=1}^{N}\rho_{a_i^k}^k(\overline{S})\right.$ and $\left.\sum_{i=1}^{N}\rho_{b_i^k}^k(\emptyset)-\sum_{i=1}^{N}\rho_{\overline{S}_i}^k(\overline{S}\setminus\overline{S}_i)\right)$ got large. However, the coefficients might decrease because of the failure rate; and the number of added cuts in which the coefficients are low might increase because of the number of scenarios. For these reasons, the tendency might be relatively weaker than it is in the deterministic problems.

We set a specific case in which the number of nodes was fixed to 50 and the number of agents was changed (five cases). The results are summarized in Table 9. The values in Table 9 are all average values from the numerical experiments when we executed 10 runs for each case. As the number of nodes increased, the computation time of Algorithm 3 increased. There were three cases (*Individual*) and four cases (*Sequential*) in which the problem could obtain (near)-optimal solutions when $N = 50$ and $M = 50$. In the instances that were not solved within 3600 s, the average ratios of $\frac{LB}{UB}$ showed 86% at both *Individual* and *Sequential*.

Figure 3 shows results for Algorithm 3 in terms of two risk levels, $\alpha = 0.01$ and $\alpha = 1$. Each node is shown as a red rectangle, and the color gets darker as the probability of event occurrences rises. When $\alpha = 1$, the result shows a decision for risk-neutral. On the other hand, when the risk level, α, is small, the result shows a decision for risk-averse. Current positions of agents and nodes before conducting Algorithm 3 are shown in Figure 3A. Figure 3B shows an initial solution given by the *Random* method. Comparing the two remaining figures (Figures 3C and 3D), agents in $\alpha = 1$ tended to cover a wider range than those in $\alpha = 0.01$. This means that it was more likely to concentrate on detecting nodes that have a high probability of event occurrence in $\alpha = 0.01$.

5.3 | Case study

A case study on detecting forest fires is conducted considering real-time data from moderate resolution imaging spectroradiometer (MODIS) provided by NASA to analyze insights of the algorithms to the problem. MODIS data are generally known to be used for dealing with forest fire monitoring.[52] The coverage of the data is worldwide, which is shown in Figure 4.[51] However, we cover only one nation (the Republic of Korea) in accordance with the applicability of the sensing problem in the case study.

In general, Fire Radiant Power (FRP) data from MODIS, which is a quantitative measure of radiant heat output, are used to estimate fire intensity.[53] This study uses normalized FRP data as the probability of event occurrences at each

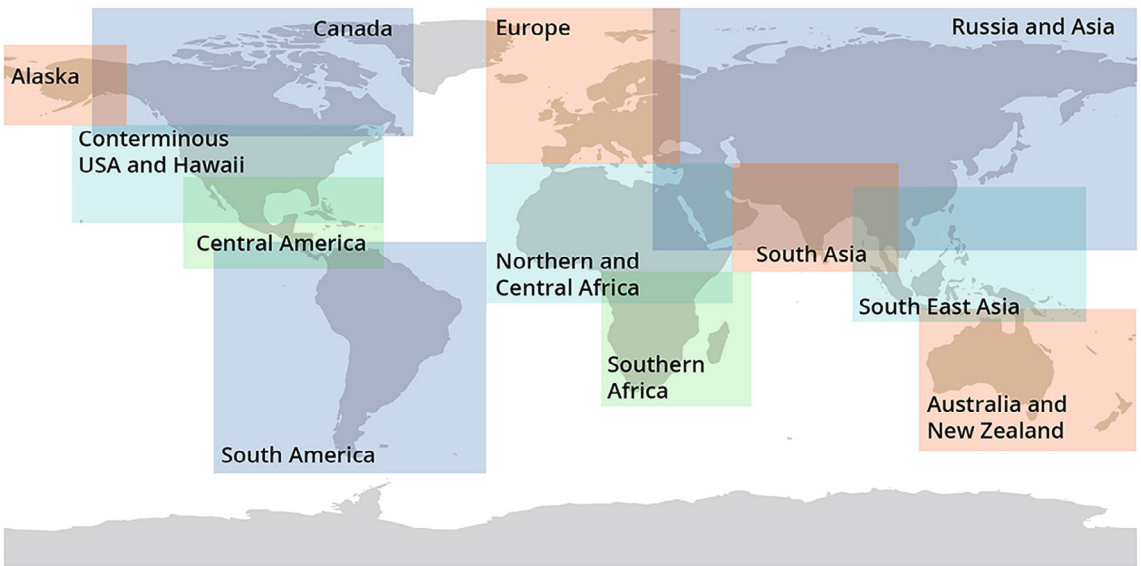


FIGURE 4 Satellite regional data coverage provided by NASA[51]

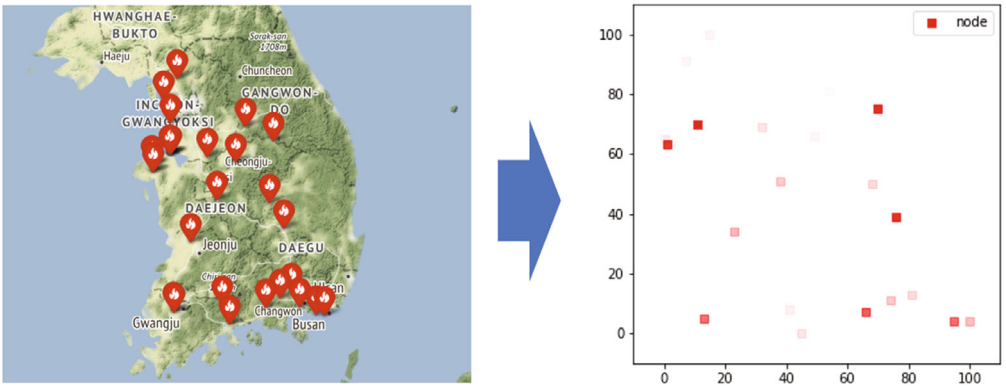
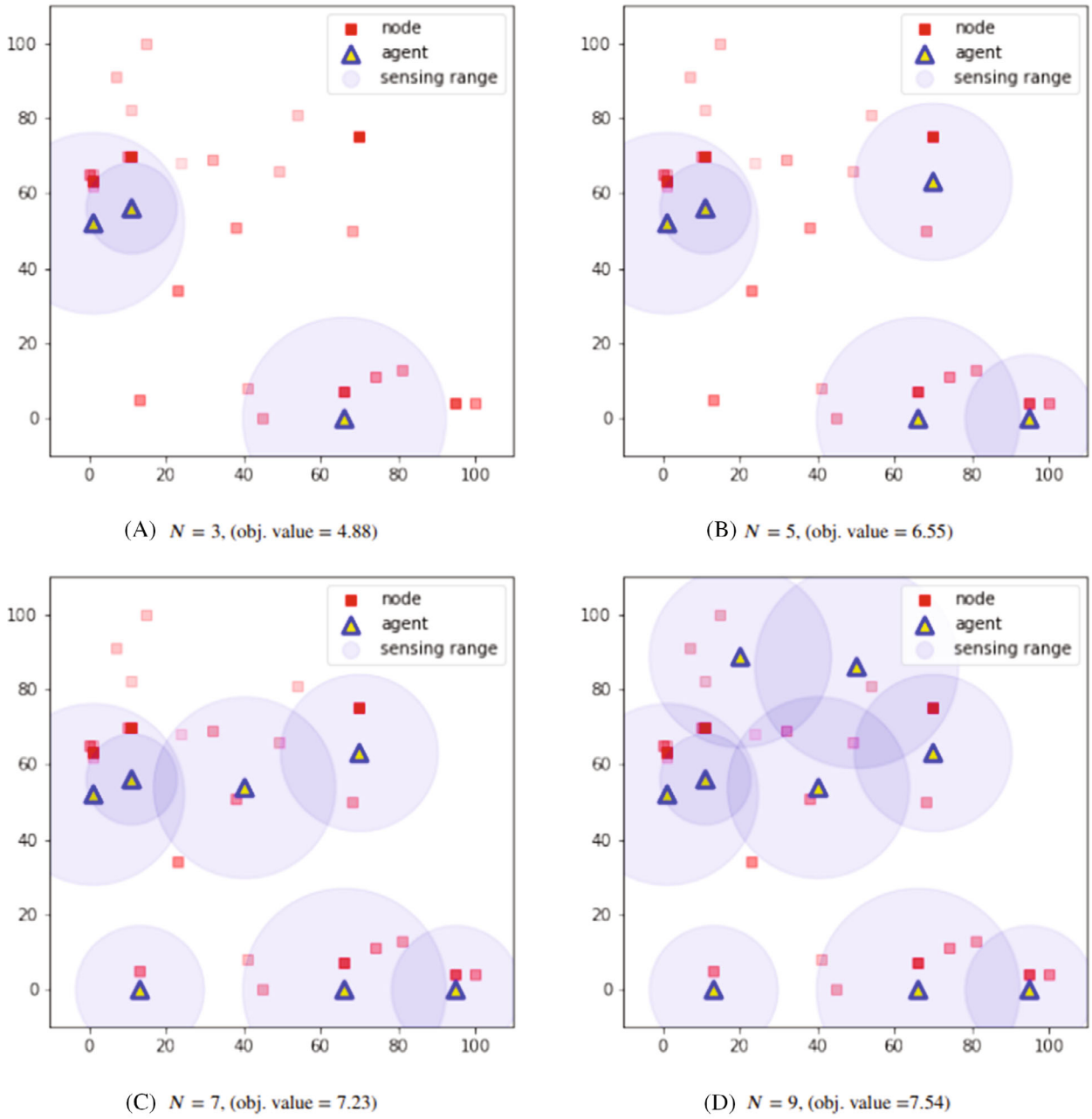


FIGURE 5 Visualization and data transformation of real-time FRP data



(A) $N = 3$, (obj. value = 4.88)

(B) $N = 5$, (obj. value = 6.55)

(C) $N = 7$, (obj. value = 7.23)

(D) $N = 9$, (obj. value =7.54)

FIGURE 6 Results of the case study according to the number of agents ($N = 3$, 5, 7, and 9)

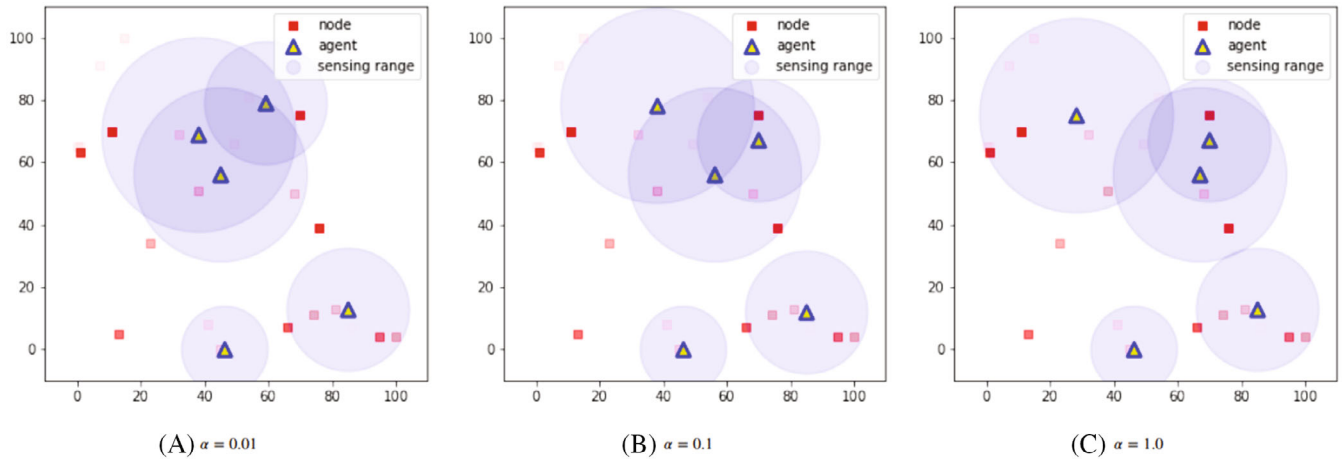


FIGURE 7 Results of the case study according to the risk levels ($\alpha = 0.01$, 0.1, and 1.0)

node. We also use geographical data in FRP data as node positions by filtering based on the latitude and longitude of the Republic of Korea. We selected the highest 30 regions among real-time FRP data as nodes. Figure 5 shows the results of visualization and data transformation. As in Figure 3, the darker each node is marked in red, the higher the FRP value, defined as the probability of event occurrence.

In the case of the deterministic problem, Figure 6 shows the strategies of agents used to maximize the objective function (1) by using Algorithm 2 according to the number of agents (3, 5, 7, and 9). As the number of agents used in this experiment increases, the objective value also increases. However, the objective values divided by the number of agents are 1.63, 1.31, 1.03, and 0.84, respectively. This means that the expected sensing effect obtained from one agent decreases as the number of agents increases. Therefore, it is necessary to decide the appropriate number of agents used considering the expected sensing effect and operating costs. We conducted numerical experiments on the risk-averse problem with five agents. Figure 7 shows the strategies of agents based on Algorithm 3 according to the three risk levels ($\alpha =$ 0.01, 0.1, and 1.0). We confirmed a tendency to cover a wider range as the risk level increases. Meanwhile, as the risk level decreases, the agents were more likely to monitor specific nodes intensively rather than to cover a wider range. This means that Algorithm 3 can provide various strategies, depending on the value of the risk level. When we deal with the sensing problem with Algorithm 3 in reality, it is important to determine effective and reasonable strategies based on the limited resources and degree of risk.

6 | CONCLUSIONS

In this article, we covered a mobile multi-agent sensing problem, which is NP-hard. It might be critical to obtain (near)-optimal solutions to reduce the probability of situations in which severe or catastrophic consequences happen. Therefore, we presented a cutting-plane algorithm for the problem to find a (near)-optimal solution efficiently. We also presented a cutting-plane algorithm for the risk-averse problem by using CVaR. We analyzed the performance of the algorithms through numerical experiments based on the three methods to set initial solutions (*Random*, *Individual*, and *Sequential*). The *Sequential* method showed better performance than the two other methods in terms of the average computation time and the ratio. This means that when the exact algorithm is used, it is also important to consider the way to set initial solutions because it affects the computation costs to find solutions. We also showed the validity and applicability of the algorithm in a case study on forest fires.

There is ample opportunity for future research on the mobile multi-agent sensing problem. First, when the severity of the event is catastrophic or the purchasing (or operating) cost of the agent is extremely high, it is important to decide the appropriate number of agents used by considering a trade-off between effectiveness and cost. An integrated model for selecting the number of agents and finding effective strategies for agents selected is needed. Other research might consider the physical limitations of the mobile agent (e.g., battery constraints or a limited number of drone stations) to reflect realistic situations.

ACKNOWLEDGMENTS

The authors are grateful for the valuable comments from the guest editors and anonymous reviewers. This research was supported by the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT and Future Planning [grant number NRF-2021R1A4A1029124] and [grant number RS-2022-00165977].

CONFLICT OF INTEREST

The authors declare no potential conflict of interests.

DATA AVAILABILITY STATEMENT

The data that was used in this study are openly available at the NASA Fire Information for Resource Management System, at https://firms.modaps.eosdis.nasa.gov/active_fire/#firms-txt.

ORCID

Ilkyeong Moon https://orcid.org/0000-0002-7072-1351

REFERENCES

1. Terelius H, Topcu U, Murray RM. Decentralized multi-agent optimization via dual decomposition. *IFAC Proc Vol*. 2011;44(1):11245-11251.
2. Yuan D, Xu S, Zhao H. Distributed primal–dual subgradient method for multiagent optimization via consensus algorithms. *IEEE Trans Syst Man Cybern B Cybern*. 2011;41(6):1715-1724.
3. Nedic A, Ozdaglar A. Distributed subgradient methods for multi-agent optimization. *IEEE Trans Automat Control*. 2009;54(1):48-61.
4. Degas A, Kaddoum E, Gleizes MP, Adreit F, Rantrua A. Cooperative multi-agent model for collision avoidance applied to air traffic management. *Eng Appl Artif Intel*. 2021;102:104286.
5. Heilig L, Lalla-Ruiz E, Voß S. port-IO: an integrative mobile cloud platform for real-time inter-terminal truck routing optimization. *Flexible Services Manufactur J*. 2017;29(3-4):504-534.
6. Jeong HY, David JY, Min BC, Lee S. The humanitarian flying warehouse. *Trans Res E Logistics Trans Rev*. 2020;136:101901.
7. Wu J, Yuan S, Ji S, Zhou G, Wang Y, Wang Z. Multi-agent system design and evaluation for collaborative wireless sensor network in large structure health monitoring. *Expert Syst Appl*. 2010;37(3):2028-2036.
8. Xia J, Jiang B, Zhang K. Resilient observer design of sensor fault estimation for discrete-time multi-agent systems: A distributed approach. *Int J Robust Nonlinear Control*. 2021;31(18):9604-9618.
9. Xu X, Zhong Y, Zhang L. Adaptive subpixel mapping based on a multiagent system for remote-sensing imagery. *IEEE Trans Geosci Remote Sens*. 2013;52(2):787-804.
10. Heyns AM, Plessis dW, Curtin KM, Kosch M, Hough G. Decision support for the selection of optimal tower site locations for early-warning wildfire detection systems in South Africa. *Int Trans Operat Res*. 2021;28(5):2299-2333.
11. Abdulsahib GM, Khalaf OI. An improved algorithm to fire detection in forest by using wireless sensor networks. *Int J Civil Eng Technol*. 2018;9(11):369-2377.
12. Mamidi S, Chang YH, Maheswaran R. *Improving building energy efficiency with a network of sensing, learning and prediction agents*. ACM; 2012:45-52.
13. Chen B, Cheng HH, Palen J. Integrating mobile agent technology with multi-agent systems for distributed traffic detection and management systems. *Trans Res C Emerg Technol*. 2009;17(1):1-10.
14. Suri D, Howell A, Schmidt D, et al. *A multi-agent architecture provides smart sensing for the nasa sensor web*. IEEE; 2007:1-9.
15. Hussein II, Stipanovic DM. Effective coverage control for mobile sensor networks with guaranteed collision avoidance. *IEEE Trans Control Syst Technol*. 2007;15(4):642-657.
16. Pavone M, Arsie A, Frazzoli E, Bullo F. Distributed algorithms for environment partitioning in mobile robotic networks. *IEEE Trans Automat Control*. 2011;56(8):1834-1848.
17. Su H, Chen G, Wang X, Lin Z. Adaptive second-order consensus of networked mobile agents with nonlinear dynamics. *Automatica*. 2011;47(2):368-375.
18. Gu K, Wang Y, Wang J, Shen Y. *On the performance analysis of cooperative detection in mobile multi-agent networks*. IEEE; 2019:1-6.
19. Liu B, Dousse O, Nain P, Towsley D. Dynamic coverage of mobile sensor networks. *IEEE Trans Parallel Distribut Syst*. 2012;24(2):301-311.
20. Sun X, Cassandras CG, Meng X. Exploiting submodularity to quantify near-optimality in multi-agent coverage problems. *Automatica*. 2019;100:349-359.
21. Hossain A, Biswas PK, Chakrabarti S. *Sensing models and its impact on network coverage in wireless sensor network*. IEEE; 2008:1-5.
22. Chen YN, Chen C. *Sensor deployment under probabilistic sensing model*. ICPS Proceedings; 2018:33-36.
23. Elfes A. Using occupancy grids for mobile robot perception and navigation. *Computer*. 1989;22(6):46-57.
24. Hossain A, Chakrabarti S, Biswas PK. Impact of sensing model on wireless sensor network coverage. *IET Wireless Sensor Syst*. 2012;2(3):272-281.
25. Henry NF, Henry ON. Wireless sensor networks based pipeline vandalisation and oil spillage monitoring and detection: main benefits for nigeria oil and gas sectors. *SIJ Trans Comput Sci Eng Appl*. 2015;3(1):1-6.

26. Yuan C, Zhang Y, Liu Z. A survey on technologies for automatic forest fire monitoring, detection, and fighting using unmanned aerial vehicles and remote sensing techniques. *Can J For Res*. 2015;45(7):783-792.
27. Clark A, Poovendran R. *A submodular optimization framework for leader selection in linear multi-agent systems*. IEEE; 2011:3614-3621.
28. Jawaid ST, Smith SL. Submodularity and greedy algorithms in sensor scheduling for linear dynamical systems. *Automatica*. 2015;61:282-288.
29. Krause A, Guestrin C. Submodularity and its applications in optimized information gathering. *ACM Trans Intell Syst Technol*. 2011;2(4):1-20.
30. Shamaiah M, Banerjee S, Vikalo H. *Greedy sensor selection: Leveraging submodularity*. IEEE; 2010:2572-2577.
31. Sun X, Cassandras CG, Meng X. *A submodularity-based approach for multi-agent optimal coverage problems*. IEEE; 2017:4082-4087.
32. Fisher ML, Nemhauser GL, Wolsey LA. *An analysis of approximations for maximizing submodular set functions—II*. Springer; 1978:73-87.
33. Nemhauser GL, Wolsey LA, Fisher ML. An analysis of approximations for maximizing submodular set functions—I. *Math Programm*. 1978;14(1):265-294.
34. Buchbinder N, Feldman M, Seffi J, Schwartz R. A tight linear time (1/2)-approximation for unconstrained submodular maximization. *SIAM J Comput*. 2015;44(5):1384-1402.
35. Conforti M, Cornuéjols G. Submodular set functions, matroids and the greedy algorithm: tight worst-case bounds and some generalizations of the Rado-Edmonds theorem. *Discrete Appl Math*. 1984;7(3):251-274.
36. Rezazadeh N, Kia SS. A sub-modular receding horizon solution for mobile multi-agent persistent monitoring. arXiv preprint, arXiv:1908.04425, 2019.
37. Wolsey LA. An analysis of the greedy algorithm for the submodular set covering problem. *Combinatorica*. 1982;2(4):385-393.
38. Lee J, Kim G, Moon I. A mobile multi-agent sensing problem with submodular functions under a partition matroid. *Comput Operat Res*. 2021;132(1):105265.
39. Qu G, Brown D, Li N. Distributed greedy algorithm for multi-agent task assignment problem with submodular utility functions. *Automatica*. 2019;105:206-215.
40. Rajaraman N, Vaze R. Submodular Maximization Under A Matroid Constraint: Asking more from an old friend, the Greedy Algorithm. arXiv preprint, arXiv:1810.12861, 2018.
41. Nemhauser GL, Wolsey LA. *Maximizing submodular set functions: formulations and analysis of algorithms*. Vol 59. Elsevier; 1981:279-301.
42. Ko CW, Lee J, Queyranne M. An exact algorithm for maximum entropy sampling. *Oper Res*. 1995;43(4):684-691.
43. Paulson JA, Mesbah A. An efficient method for stochastic optimal control with joint chance constraints for nonlinear systems. *Int J Robust Nonlinear Control*. 2019;29(15):5017-5037.
44. Kawahara Y, Nagano K, Tsuda K, Bilmes JA. Submodularity cuts and applications. 2009:916-924.
45. Maehara T. Risk averse submodular utility maximization. *Operat Res Lett*. 2015;43(5):526-529.
46. Wu HH, Kucukyavuz S. On a Class of Risk-averse Submodular Maximization Problems. arXiv preprint, arXiv:1903.08318, 2019.
47. Zhou L, Tokekar P. An approximation algorithm for risk-averse submodular optimization. arXiv preprint, arXiv:1807.09358, 2018.
48. Wu HH, Küçükyavuz S. An exact method for constrained maximization of the conditional value-at-risk of a class of stochastic submodular functions. *Operat Res Lett*. 2020;48(3):356-361.
49. Artzner P, Delbaen F, Eber JM, Heath D. Coherent measures of risk. *Mathe Ffinance*. 1999;9(3):203-228.
50. Rockafellar RT, Uryasev S. Optimization of conditional value-at-risk. *J Risk*. 2000;2:21-42.
51. NASA Fire Information for Resource Management System. Accessed July, 2022. https://firms.modaps.eosdis.nasa.gov/active_fire/#firms-txt.
52. Hua L, Shao G. The progress of operational forest fire monitoring with infrared remote sensing. *J Forestry Res*. 2017;28(2):215-229.
53. Yao J, Raffuse SM, Brauer M, et al. Predicting the minimum height of forest fire smoke within the atmosphere using machine learning and data from the CALIPSO satellite. *Remote Sens Environ*. 2018;206:98-106.

How to cite this article: Kim G, Lee J, Moon I. An exact solution approach for the mobile multi-agent sensing problem. *Int J Robust Nonlinear Control*. 2023;33(17):10405-10424. doi: 10.1002/rnc.6510