Computers and Operations Research 119 (2020) 104936

Contents lists available at ScienceDirect

Computers and Operations Research

journal homepage: www.elsevier.com/locate/cor

Unmanned aerial vehicle set covering problem considering fixed-radius coverage constraint

Youngsoo Park[a], Peter Nielsen[b], Ilkyeong Moon[a,c,*]

[a] Department of Industrial Engineering, Seoul National University, Seoul 08826, Republic of Korea
[b] Department of Materials and Production, Aalborg University, Aalborg 9220, Denmark
[c] Institute for Industrial Systems Innovation, Seoul National University, Seoul 08826, Republic of Korea

ARTICLE INFO

Article history:
Received 2 May 2019
Revised 2 March 2020
Accepted 2 March 2020
Available online 5 March 2020

Keywords:
Disaster management
Emergency wireless network
Unmanned aerial vehicle
Set covering problem
Branch-and-price

ABSTRACT

This paper models the problem of providing an unmanned aerial vehicle (UAV)-based wireless network in a disaster area as a set covering problem that takes into consideration the operational constraints and benefits of UAVs. The research presents a branch-and-price algorithm and two approximation models of the quadratic coverage radius constraint in a simple discretization and a linear pairwise-conflict constraint based on Jung's theorem. In computational experiments, we found that the exact branch-and-price algorithm and two approximation models are applicable for realistic-scaled problems with up to 100 demand points and 2,000 m of coverage radius.

1. Introduction

Various fields of industry are showing an increasing interest in unmanned aerial vehicle (UAV). The main advantage of UAVs is their autonomous swarm operation capability (Kim et al., 2018; Kim and Moon, 2019), which enables the system to execute multiple tasks simultaneously at a low cost and without human intervention. Under the most recent positioning system, which guarantees precise location recognition, UAVs can be operated as flexible, expendable resources in diverse industries and environments. For example, UAVs have been used not only in the military, surveillance, and logistics applications but also in disaster management, including casualty search and relief logistics (Chowdhury et al., 2017). An emerging approach is the use of UAVs to establish an emergency wireless network in a disaster area. In a natural disaster situation with mass destruction over a large area, such as an earthquake, tsunami, or flood, the damage to infrastructure facilities often leads to immediate and secondary casualties. Survivors in the disaster area who cannot evacuate immediately require a means of communication with the outside world. At the same time, authorities require a system for monitoring survivors and the scene of the disaster.

Many cases have shown that in disaster areas, the wireless network is one of the systems to be recovered first. Survivors of disasters often use wireless networks to inform the authorities and their relatives the status they are encountered. For example, some survivors of the 2011 Great East Japan Earthquake watched for updates of the disaster and posted about their situation on Twitter and Facebook (Wallop, 2011). Besides, some survivors used the wireless network to inform authorities of the current immediate situation and to request rescue (Aida et al., 2013). Similarly, survivors of the 2010 Haiti earthquake (Heinzelman and Waters, 2010) and Hurricane Harvey in southeast Texas (Holley, 2017) used Twitter to request evacuation, which allowed authorities and homegrown volunteer groups to provide aid.

Efforts have also been underway to develop systems to monitor disaster scenes. After the rapid growth of sensor technology, the only remaining hurdle is to maintain connectivity between the disaster scene and authorities. Thus, researchers in both academia and industry are helping to develop UAV-based wireless networks that can help recover a temporary wireless connection in a disaster environment. One issue with a UAV-enabled wireless network is ensuring that the overall system (including the sensors and the UAV) has enough power to maintain operation for the time needed. Thus, researchers are searching for ways to minimize energy consumption and maximize the overall system lifetime. The UAV routing model can be used for similar problems,

* Corresponding author at: Department of Industrial Engineering, Industrial Engineering & Institute for Industrial Systems Innovation, Seoul 08826, Republic of Korea.
E-mail addresses: simulacrum@snu.ac.kr (Y. Park), peter@mp.aau.dk (P. Nielsen), ikmoon@snu.ac.kr (I. Moon).

https://doi.org/10.1016/j.cor.2020.104936
0305-0548/

such as a visual surveillance and monitoring system based on camera-mounted UAVs that need to operate near the actual site. Ho et al. (2015) used particle swarm optimization as an approximation algorithm to optimize the UAV trajectory along waypoint candidates. Zhan et al. (2018) synchronized the wake-up schedule of sensors and the UAV trajectory. Zeng et al. (2016) and Wu et al. (2018) maximized the throughput controlling trajectory and speed of the UAV. Zeng et al. (2018) approximated the makespan-minimization problem as a generalized traveling salesperson problem and proposed a two-stage algorithm to solve it.

Another issue is the UAVs' need to hover in a planned area to maintain continuous network connectivity, which is addressed in this paper. There are three closely related approaches to the continuous network and facility location problems. The first approach is the continuous location problem. The continuous location problem, including the minimal covering circle problem, aims to decide the position of a facility in the xy-plane with various types of distance (e.g., Euclidean, rectangular, and p-norms). While deciding the position of one facility and its coverage radius, a weighted sum or a minimax distance is used as an objective. Accordingly, every facility and demand point pair is considered without division. Drezner et al. (2001) and Plastria (2001) provide detailed information about the continuous location problem approach.

The second approach is the clustering problem. Some of the considerable research conducted on the clustering approach has focused on K-means (Periyasamy et al., 2016; Sasikumar and Khara, 2012), modified K-means (Periyasamy et al., 2016), disk covering (Mozaffari et al., 2016b), and circle packing in a circle (Mozaffari et al., 2016a). In the field of facility location, there are similar problems of p-center (Daskin and Maass, 2015) and p-cover (Calik et al., 2015). These problems consider the given number of the clusters K (or p) as the primary constraint instead of the physical limitation of the coverage radius. Existing studies use a function (e.g., sum or minimax) of various types of distances as an objective because the purpose of traditional clustering problems is not the physical clustering but the classification of data.

The third approach is the set covering problem. Facility location problems with a set covering approach make the decision using given candidates with predefined positions for facility locations. Because every location of the facility is already known, the constraint on the coverage radius is easily considered by checking the feasibility of each facility and demand point pair. Chandrashekar et al. (2004) modeled a two-layer network that consists of mobile ad hoc networks (MANET) and a covering UAV network. Zorbas et al. (2016) proposed a minimum-cost drone location problem that takes into consideration the network coverage changing over flight altitudes.

The UAV set covering problem (USCP) has two distinct characteristics that the approaches mentioned above do not have. First, the objective of the USCP is to optimize the cardinality of the UAVs, whereas the continuous location and clustering problem approaches minimize the cost function, which is usually related to the arcs of the network. The arc-related cost function considers the relation between every facility and demand point pair, which does not require an extra decision on a set partition. Interested readers are referred to Boonmee et al. (2017) for a detailed literature review. For the USCP, changing the objective significantly reduces the ability to resolve the problem. Without a decision on a set partition, the linear relaxation bound for the USCP does not provide any information, as explained in Section 3.1. Note that the solution algorithm of p-center problem is exponential in p (Capoyleas et al., 1991). Because the cardinality of the UAV is not predefined as a parameter in the USCP, the researcher must iterate over p up to the number of demand points, which will grow exponentially.

Second, the USCP brings the positions of facilities into the decision problem. The traditional set covering problem approach decides among given candidates for facility locations. This is because a real disaster situation has a wide variety of constraints and a handful of possible sites for facility locations. Also, to avoid impractical solution algorithms (Toregas et al., 1971), models make decisions among a limited number of candidates. When the facility candidate is predefined, the availability between each demand point and each candidate facility location is defined as well; thus, it is straightforward to apply coverage radius as a constraint. For literature focusing on coverage constraints, we refer readers to ambulance location and relocation problems (Ahmadi-Javid et al., 2017; Aringhieri et al., 2017; Bélanger et al., 2019; Brotcorne et al., 2003). However, in the case of the USCP, there is flexibility to position the UAV freely on the xy-plane. This imposes the problem of having to check every possible subset of demand points, which will grow exponentially.

Therefore, it is difficult to apply the knowledge from existing studies to the USCP straightforwardly. Despite the straightforward definition of the problem and the model, which will be presented later, the USCP suffers from a computational burden; however, the necessity of the research has recently begun to emerge. The introduction of UAVs into the set covering problem has changed the situation dramatically because of the flexible positioning and the limitations of network coverage significantly affect the model. We noticed a gap between the related approaches and the technologies required at the scene of a disaster. Thus, this paper presents a set covering problem without predefined candidate positions and with the consideration of a fixed-radius coverage constraint to fill the gap mentioned above.

The wireless network is assumed to be uncapacitated to show and maximize the effects of the disaster environment's topographic structures. To utilize the knowledge of the topographic structure for the solution algorithm, we developed a branch-and-price (B&P) algorithm for the USCP as other research considering assignment constraints (e.g., set covering, clique covering, and vehicle routing; Ji and Mitchell, 2007; Johnson et al., 1993; Vance, 1998; Vance et al., 1994). The B&P algorithm runs a column generation (CG) method on every node in a branch-and-bound (B&B) tree, defining new patterns of the demand points to be covered by a UAV as new variables. The reformulation associated with the B&P algorithm strengthens the linear programming relaxation (LP) bound and decreases symmetries in the branching tree. It enabled the proposed B&P algorithm to provide the optimal solution in a reasonable timescale for both a small-sized artificial dataset and a realistic-scale dataset in computational experiments.

Even though the USCP is reformulated, the mixed-integer quadratic coverage constraint remains in the CG subproblem. An approximation model is designed to avoid the numerical instability incurred by the coverage constraint and to improve the computation speed. In the approximation model, a linearized, pairwise-conflict constraint based on the sufficient condition substitutes the quadratic constraint. We observed that the B&P algorithm for the pairwise-conflict constraint approximated model is numerically stable and provides a time-efficient solution with applicable optimality.

The rest of this paper is structured as follows: Section 2 proposes the problem description and the mathematical model of the standard formulation. A direct way to discretely approximate the mixed-integer quadratic constraint into the integer constraint is also proposed. Section 3 describes, in detail, the B&P approach for the USCP and the pairwise-conflict constraint approximation based on Jung's theorem. This section also presents the comparison between two approximation models and the overall algorithmic framework for further clarification. Section 4 presents the computational experiments conducted, including the algorithmic performances of four proposed models. In this section, we analyze the managerial insights for practical applica-

tions in a disaster environment. Finally, Section 5 concludes the research.

2. Problem definition

This section presents a detailed description of the USCP. UAVs construct a wireless network to restore connectivity for survivors in a disaster area. The ultimate scope of the UAV operation problem should be the development of an optimal flight schedule for an overall UAV system within the constraints of battery capacity. The interval scheduling or interval partitioning problem expands the set covering problem to the time dimension. The flight time to the target position and the duration of hovering, considering the battery capacity of the UAVs, would be incorporated into the interval scheduling problem. However, this problem is simplified in the set covering problem to allow more explicit consideration of the UAV system's characteristics. The set covering approach provides a less dimensionally complex and more intuitive solution as compared with the scheduling approach; this would allow authorities to manage the system efficiently. The proposed USCP provides a rough-cut response plan that can be used in creating a routing and scheduling plan. The solution for the USCP, which consists of the position and the assignment of demand points for each UAV, can be used as a set of feasible tasks. For this reason, the USCP minimizes the number of UAVs needed to cover every demand points rather than maximizing the number of demand points covered by a limited number of UAVs.

2.1. Problem description

The cardinality minimization problem is designed for the immediate response phase after a disaster. Thus, authorities can obtain information on the approximate number of UAVs to create a rough-cut response plan. The cardinality minimization problem can be transformed into the coverage maximization problem, which maximizes the number of demand points covered by a given number of UAVs. Since the hard target in disaster management is to minimize the damage of human life through the utmost efforts, the constraint is set to fulfill every demand.

The assumptions of the presented problem are defined as follows:

1. The information on the positions of demand points is already known.
2. Each UAV has an identical coverage distance.
3. There is no restriction on a UAV's hovering position in the xy-plane.
4. A demand point is covered if it is in the coverage circle.
5. There is no transmission capacity limitation on the wireless network.
6. There is no overlap interference between UAVs or shadowing effect incurred by buildings.

It is assumed to have initial information on the positions of demand points, which can be acquired by a primary search or by experts of disaster management. If there is a solution algorithm to solve the USCP efficiently, authorities can provide a more advanced response through optimizing the model iteratively and adding new information. To utilize the UAV's full capability, there is no restriction on its position; in other words, there is no predefined candidate for each UAV to fly. This research focuses on the geometrical coverage constraint rather than the capacity constraints of the wireless network. Each demand point is *covered* by a UAV if it is in the coverage circle, regardless of a UAV's capacity on the network's accessor or transmission. The characteristic that authorities and survivors both have limitations on resources grounds the assumption. For the authorities, there are limitations on the number of UAVs to be invested. At the same time, survivors conserve their battery of the mobile devices as much as possible because there is a lack of assurance of rescue. As a result, access to the wireless network only occurs if absolutely necessary, which makes capacity constraints of the network immaterial.

Fig. 1 presents an overview of the USCP. Under the given information of the static position of demand point, the objective of the UAV set covering problem considered in this paper is to minimize the number of UAVs required to cover every demand point in a disaster situation. UAVs can be located without any restriction on an xy-plane. The network-covered area is defined by the employment and position of UAVs, and a demand point is covered if and only if it is inside the area.

2.2. Mathematical formulation

Based on the problem defined in Section 2.1, a mathematical model is developed. The followings are the notations used in the standard mathematical formulation for the USCP:

Set

N set of demand points.

Parameters

a_i^x position of demand point i on x-coordinate. $\forall j \in N$
a_i^y position of demand point i on y-coordinate. $\forall j \in N$
R coverage radius of a UAV.

Decision variables

$y_j = \begin{cases} 1, & \text{if UAV } j \text{ is used.} \\ 0, & \text{otherwise.} \end{cases}$ $\forall j \in N$

$x_{ij} = \begin{cases} 1, & \text{if demand point } i \text{ is covered by UAV } j. \\ 0, & \text{otherwise.} \end{cases}$ $\forall i \in N$ $\forall j \in N$

$c_j^x \in \mathbb{R}$, position of UAV j on x-coordinate. $\forall j \in N$
$c_j^y \in \mathbb{R}$, position of UAV j on y-coordinate. $\forall j \in N$

The set N consists of the demand points and represents the survivors in the disaster area. Positions of demand points and the coverage radius of a UAV are given as parameters. There are two types of decision variables: binary decision variables related to the location-allocation problem and position decision variables on the xy-plane. UAV j and y_j are predefined for each demand point to cover every extreme case. The following is the standard mathematical formulation of the USCP. For distinction, the formulation will be renamed as Euclidean standard (ES) formulation.

$$\min \quad \sum_{j \in N} y_j \tag{1}$$

$$\text{s.t.} \quad x_{ij} \le y_j, \qquad \forall i \in N, \forall j \in N \tag{2}$$

$$\sum_{j \in N} x_{ij} \ge 1, \qquad \forall i \in N \tag{3}$$

$$(a_i^x - c_j^x)^2 + (a_i^y - c_j^y)^2 \le R^2 + M(1 - x_{ij}), \qquad \forall i \in N, \forall j \in N \tag{4}$$

$$x_{ij} \in \{0, 1\}, \qquad \forall i \in N, \forall j \in N \tag{5}$$

$$y_j \in \{0, 1\}, \qquad \forall j \in N \tag{6}$$

$$c_j^x, c_j^y \in \mathbb{R}, \qquad \forall j \in N \tag{7}$$

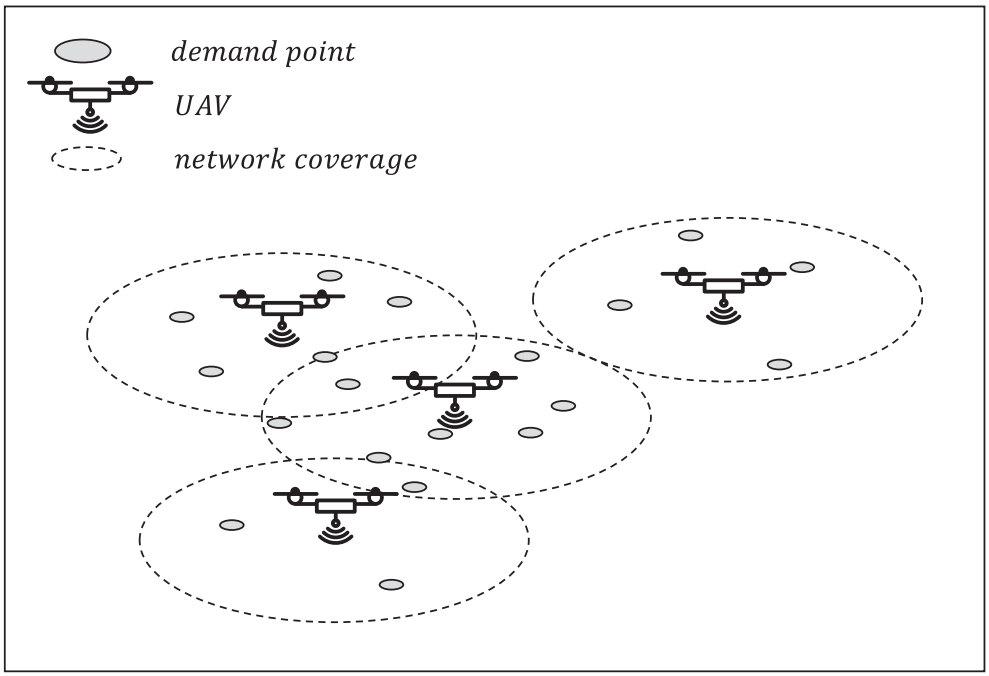


Fig. 1. Overview of the USCP.

The objective of the mixed-integer quadratically-constrained programming (MIQCP) model is to minimize the total number of UAVs used to cover the demand points. Constraint (2) is a linking constraint between a demand point and a UAV; the deployment of a UAV precedes the assignment of a demand point. Constraint (3) is a demand assignment constraint; every demand is required to be fulfilled by at least one UAV. Constraint (4) is a mixed integer quadratic constraint that relates the position-coverage of UAV and its usage. A demand point i is covered by a UAV j only if the distance between the position of the demand point i, (a_i^x, a_i^y) and the position of UAV j, (c_j^x, c_j^y) is less than the coverage radius R. Constraints (5)–(7) define the dimension of the decision variables. The quadratic shape of Constraint (4) originates in Comley (1995). It is hard to solve the ES model within an applicable time, even for a small-sized problem. To tackle the intractability of the ES model, a natural approximation model based on discretization is proposed in the next section.

2.3. Discrete approximation model

Constraint (4) is quadratic because of the continuous decision variable for the position of the UAV, c_j^x and c_j^y. The simplest approximation of the ES model to linearize the quadratic constraint is to discretize the xy-plane into grids and consider every lattice point as a candidate for the position of a UAV. The following are the new set and parameters used in the mathematical formulation of the discrete approximation (DA) model:

Set

M set of candidates of flight position of UAV.

Parameters

b_j^x	flight position of UAV j on x-coordinate.	$\forall j \in M$
b_j^y	flight position of UAV j on y-coordinate.	$\forall j \in M$
α_{ij}	binary feasibility of UAV j to cover demand point i.	$\forall i \in N$, $\forall j \in M$

Decision variables

$$y_j = \begin{cases} 1, & \text{if UAV } j \text{ is used.} \\ 0, & \text{otherwise.} \end{cases} \quad \forall j \in M$$

$$x_{ij} = \begin{cases} 1, & \text{if demand point } i \text{ is covered by UAV } j. \\ 0, & \text{otherwise.} \end{cases} \quad \forall i \in N, \forall j \in M$$

Unlike in the ES model, UAV $j \in M$ is predefined for each lattice point on the xy-plane, separated into grids. The binary feasibility α_{ij} is defined based on the distance between the demand point and the flight position of UAV j. α_{ij} equals 1 if $\sqrt{(a_i^x - b_j^x)^2 + (a_i^y - b_j^y)^2} \leq R$ and 0 otherwise. Except for the domain of the binary decision variable y_j and the discretization of the continuous decision variables c_j^x and c_j^y, the mathematical formulation of the DA model is almost identical to that of the ES model:

$$\min \quad \sum_{j \in M} y_j \tag{8}$$

$$\text{s.t.} \quad x_{ij} \leq y_j, \qquad \forall i \in N, \forall j \in M \tag{9}$$

$$\sum_{j \in M} \alpha_{ij} x_{ij} \geq 1, \qquad \forall i \in N \tag{10}$$

$$x_{ij} \in \{0, 1\}, \qquad \forall i \in N, \forall j \in M \tag{11}$$

$$y_j \in \{0, 1\}, \qquad \forall j \in M \tag{12}$$

Set M is defined based on the size of the grid and the boundary of the demand points. The extreme values of the leftmost, rightmost, uppermost, and lowermost points become the boundary of set M. The smaller each grid is, the more precise the approximation becomes, but at the same time, the size of set M increases quickly in squares. Even though the approximation of the quadratic constraint accelerated the computation speed, the size of the problem in terms of decision variables and constraints can be excessively large. Thus, a scientific criteria to identify an efficient grid size is vital to implement the DA model. The detailed performance

of the DA model and the criteria for the grid size is analyzed in Sections 3.4 and 4.

3. Branch-and-price approach for the USCP

Section 3.1 presents an extended formulation for the B&P algorithm. Section 3.2 introduces a detailed branching strategy related to the B&P algorithm on the USCP. Section 3.3 examines Jung's theorem and the approximation model with pairwise-conflict constraints based on Jung's theorem. Section 3.4 compares two approximation models based on the approximation ratio. Section 3.5 presents the overall algorithmic framework to use pairwise-conflict constraint approximation model in a disaster situation.

3.1. An extended formulation of the USCP

To utilize the structural knowledge of the problem's feasible solution, we reformulated the ES into the extended formulation. One strong point of the nominal set covering problem is the small integrality gap and the tendency for the LP relaxation to provide an integer solution (Simchi-Levi et al., 2005). However, the LP relaxation of the USCP neutralizes the coverage constraint. Unlike in the nominal set covering problem, the coverage constraint is considered jointly by Constraints (3) and (4) in the ES formulation. The LP relaxation separates the relation among x_{ij}, c_j^x, and c_j^y. Thus, the solution of the LP relaxation does not satisfy the coverage constraint. Moreover, because the objective of the USCP is to minimize the number of homogeneous UAV without a consideration of capacity, the relaxation always provides the LP bound as 1. In the extended formulation, the fixed-radius coverage constraint with Euclidean distance (4) is considered implicitly in the decision variable; therefore, the solutions of the LP relaxation satisfy the coverage constraints, which obtains tighter LP bounds than in the nominal set covering problem. Another advantage of the decomposition was the elimination of symmetry among solutions, which obstructed the search on the B&B algorithm (Vanderbeck and Wolsey, 2010).

Each column in the extended formulation defines a set of demand points that can be covered by one UAV. In this approach, one makes the *column-wise* decision instead of the UAV - demand point pair decision by choosing to use particular columns and cover the included demand points. For example, if columns $j_1 = \{1, 3\}$, $j_2 = \{2, 3\}$, and $j_3 = \{1, 2\}$ are considered, one can cover demand points {1, 2, 3} by selecting columns j_1 and j_2, or j_2 and j_3. Let Ω be the set of every feasible column that implicitly represents a set of demand points covered by one UAV. For each assignment plan $j \in \Omega$, the inclusion of each demand point i is defined as a binary parameter w_{ij}. A binary decision variable z_j is defined for each feasible column to denote the adoption. The extended formulation model of the USCP is represented in the following integer program:

$$\min \quad \sum_{j\in\Omega} z_j \tag{13}$$

$$\text{s.t.} \quad \sum_{j\in\Omega} w_{ij} z_j \geq 1, \qquad \forall i \in N \tag{14}$$

$$z_j \in \{0, 1\}, \qquad \forall j \in \Omega \tag{15}$$

Objective function (13) minimizes the cardinality of UAVs operated to cover every demand point. Constraint (14) is a demand assignment constraint. For each demand point, at least one active assignment plan is required to cover it. In the extended formulation, it is impossible to define intact Ω and every decision variable z_j because the size of the set Ω is exponential on the number of demand points m. The optimality under the current basis is verified by a subproblem called a *pricing subproblem*, which identifies a new column for entering the basis to improve the solution; the operation is iterated until no new column with a negative reduced cost is found. The B&P algorithm is a B&B algorithm with a CG technique implemented at each node. The branching occurs when the LP solution after the CG terminates does not satisfy the integrality. Let π_i be a dual price associated with constraint (14); additional columns for the restricted master problem can be generated by solving the following pricing problem:

Decision variables

$$x_i = \begin{cases} 1, & \text{if demand point } i \text{ is covered by the generated column.} \\ 0, & \text{otherwise.} \end{cases} \quad \forall i \in N$$

$c^x \in \mathbb{R}$, position of UAV of the generated column on x-coordinate.

$c^y \in \mathbb{R}$, position of UAV of the generated column on y-coordinate.

$$\min \quad 1 - \sum_{i=1}^{m} \pi_i x_i \tag{16}$$

$$\text{s.t.} \quad (a_i^x - c^x)^2 + (a_i^y - c^y)^2 \leq R^2 + M(1 - x_i), \qquad \forall i \in N \tag{17}$$

$$x_i \in \{0, 1\}, \qquad \forall i \in N \tag{18}$$

$$c^x, c^y \in \mathbb{R}, \tag{19}$$

The pricing subproblem for the CG is equivalent to the Lagrangian subproblem of the ES formulation. Because all the UAVs are assumed to be identical, the cost parameter for using a UAV j is set to be 1 for every UAV in Formulation (1). Thus, the pricing subproblem is identical for every UAV. The objective function (16) calculates the cost to employ one UAV, as we must always use one UAV. Moreover, the dual price π_i subtracts the covering effect of the demand point. The fixed-radius coverage constraint with Euclidean distance, Constraint (4) in the ES formulation, is considered in Constraint (17), which provides the feasible column to be covered by one UAV. The B&P algorithm over extended formulation is renamed as Euclidean branch-and-price (EBP). For the initial restricted master problem, we assigned a UAV to each demand point. In other words, each initial column covered one demand point, and the number of initial columns was the same as the number of demand points. In this way, the initial restricted master problem had a feasible LP relaxation and could provide dual values, which were used in the pricing problem. Powerful heuristic algorithms exist to cluster demand points efficiently, and one can easily use these algorithms to provide initial columns while implementing the system in the real application.

3.2. Branching strategies

Branching is required when the CG terminates and the optimal solution does not satisfy integrality. New constraints are added by the branching to divide the solution space without losing any feasible solution and to gain the optimal integer solution. The branching decision is based on the standard formulation rather than on an extended (disaggregated) formulation, because branching on the decision variable causes an unbalance in the branch-and-bound tree and requires massive modifications in the pricing subproblem (Desaulniers et al., 2006). The Ryan–Foster branching rule (Ryan and Foster, 1981) is often used in the set partitioning problem (Ji and Mitchell, 2005). In this rule, the branching decision controls whether two demand points are simultaneously covered by a UAV or not. It is modeled by fixing the coexistence of decision variables x_i and x_k for subproblem, which represents the assignment of demand points i and k for each UAV. In detail, we

can identify a pair of the most fractured demand points based on the solution from extended formulation. Because the CG is operated on the restricted master linear program (RMLP), the employment of each column is given as a fractional value. Based on the fractional solution of the RMLP, the degree of coexistence of a pair of demand points v_{ik} is calculated. For each pair of demand points, the value of the fractional solution of a column $\tilde{z}_j$ is aggregated if both demand points are included:

$$v_{ik} := \sum_{j\in\Omega, w_{ij}=w_{kj}=1} \tilde{z}_j.$$

The pair of demand points with the degree of coexistence nearest to 0.5 is chosen for the branching.

When the branching is executed, division of feasible solutions is required in both the master problem and the pricing subproblem. In the master problem, the existing columns should be divided into two groups based on the coexistence of the pair of demand points chosen for the branching. This separates the columns covering both demand points into one branch and the columns covering only one demand point of the pair into another branch. In the pricing subproblem, a new pairwise-conflict constraint is added to enforce the acceptance or prohibition of the coexistence. Because the subproblem makes decision of the demand points to be covered, the addition of the pairwise-conflict constraint does not change the structure of the problem.

3.3. Pairwise-conflict constraint approximation model based on Jung's theorem

The reformulation provides a better LP bound and eliminates the symmetries in the branching tree. Furthermore, the advancement of the nonlinear solver engine enables the solution algorithm to solve a knapsack-like pricing subproblem efficiently regardless of the quadratic constraints. In ES and extended formulations of the USCP, quadratic constraints (4) and (17) represent the coverage circle around each UAV. Even though the commercial solver can find the optimal solution of USCP within the appropriate time, it is still necessary to find a more practical model. One reason for this is the numerical instability of the coverage constraint. The binary decision variable x_i and the continuous decision variables c^x and c^y coexist in Constraint (17). Because the scale of the coverage distance of one UAV and overall xy-plane is compared in one inequality in quadratic form, R^2 and M can easily have the difference of 10^8 units when 10^{-8} is the limit of the solver's feasibility tolerance.

For the same reason, it is not possible for the solver to use the built-in presolver and heuristics, since they can provide infeasible solutions. In addition, linear constraints are usually preferred over nonlinear constraints because the movement between feasible points is more straightforward when the solution space is linear rather than curved (Elzinga et al., 1976). Therefore, in most cases, linearization can accelerate the computation speed.

In the field of geometry, finding the minimum enclosing ball of a set has been an important question. Under the given set, Welzl (1991) proposed a randomized linear programming algorithm that runs in linear time. Because a circle is defined by three points, Welzl's algorithm recursively chooses three points from the given set of points to find the enclosing circle. However, due to the recursive characteristic, it is not possible to apply Welzl's algorithm as a constraint in the mathematical model.

Instead of the minimum enclosing ball approach, which cannot be used as a constraint, a sufficient condition can be used for the approximation. If every pair in the set of demand points satisfies the linearized conflict constraint, the sufficient condition ensures that the set will be covered together in one circle. The conflict constraint is widely used in optimization models. Sadykov and Vanderbeck (2012), Gendreau et al. (2016), and Manerba and Mansini (2016) used conflict constraints to model predefined incompatibility between choices. Grötschel and Wakabayashi (1989), Hoffman and Padberg (1993), and Borndörfer and Weismantel (2000) developed valid inequalities for the solution algorithm. In our approximation, a pairwise-conflict constraint inspired by the Ryan–Foster branching strategy and Jung's theorem is used to identify the pairs of demand points that can coexist within a given coverage distance.

Jung (1901) proposed an inequality between the diameter and the radius of the minimum enclosing ball of a set:

Theorem 1 (Jung's theorem)**.** *Considering a compact set $K \subset \mathbb{R}^n$ and let the diameter of a set K as $d(K) := \max_{p,q\in K} ||p-q||_2$. There exists a closed ball with radius*

$$r \le d(K)\sqrt{\frac{n}{2(n+1)}}$$

that contains K.

In the case of the xy-plane ($n = 2$), according to Jung's theorem, a circle with $r \le \frac{d(K)}{\sqrt{3}}$ containing the given compact set K exists. However, we can distinguish a sufficient condition for some sets to be enclosed in a closed ball under the given radius R.

Lemma 2. *Considering a compact set $K \subset \mathbb{R}^2$. For a given $R \in \mathbb{R}^1$, if $d(K) \le \sqrt{3}R$, then there exists a closed ball with radius $r \le R$.*

According to Lemma 2, if the coverage radius R is given and the distance of every pair of demand points in a set K is smaller than $\sqrt{3}R$, the set K can be covered by one circle. The approximation of Constraints (4) and (17) are modeled as pairwise-conflict constraints. A constraint on the diameter of a set K can be substituted by a set of constraints that includes the constraint of distance between every pair of demand points. Let d_{ik} be the distance between demand points i and k. Constraints (4) and (17) can be approximated into Constraints (20) and (21), respectively:

$$x_{ij} + x_{kj} - 2 + C_{ik} \le 0, \qquad \forall i, j, k \in N \tag{20}$$

$$x_i + x_k - 2 + C_{ik} \le 0, \qquad \forall i, k \in N \tag{21}$$

where pairwise-conflict parameter $C_{ik} := \dfrac{(d_{ik})^2 - 3R^2}{(\max_{i,k\in N}[d_{ik}])^2 - 3R^2}$. If $d_{ik} \le R\sqrt{3}$, Constraint (20) and (21) become redundant. Otherwise, $0 \le C_{ik} \le 1$ and Constraint (20) and (21) define pairwise-conflict constraints. The model is named as pairwise-conflict constraint approximation (PCA) model. For distinction, the approximated formulations are renamed as pairwise-conflict constraint approximated standard formulation (PCS) and pairwise-conflict constraint approximated branch-and-price algorithm (PCBP). Let $\boldsymbol{x}, \boldsymbol{y}$ be a feasible solution of a PCS model. According to Lemma 2, $\boldsymbol{x}, \boldsymbol{y}$ is also feasible for the ES model. A feasible solution to a pricing subproblem of PCBP is likewise feasible for the pricing subproblem of EBP. Note that the approximations of the coverage constraints have the same structure as the branching constraints. Moreover, this means that the structure of the problem does not need to be changed while executing branching over the original model.

PCBP has the same extended formulation as EBP, and the approximated CG subproblem has the same objective function as Formulation (16) and Constraints (18) and (19). Constraint (21) replaces Constraint (17) to represent the approximated coverage constraint. In the CG subproblem, Constraint (21) requires that every pair of demand point chosen for a new column must be within a certain distance. After the approximation, the decisions of the UAV position and the set partition are decomposed, and only the decision of the set partition becomes relevant. In other words, when

we approximate the geometric network structure, the pairwise-conflict constraint can work as a filter that abstracts the original network into the digitized network. Connectivity between a pair of demand points remains only if Jung's theorem guarantees the co-existence. After the abstraction, it is not necessary to consider the length of each arc between the demand points. By the approximation, the physical distribution of demand points is abstracted to a network that only considers the pairwise connectivity between demand points. In the abstracted network, the master problem is to cover every demand point with the minimum number of cliques, and the pricing subproblem is translated into the category of finding the maximum vertex clique.

The clique partitioning problem is one of the most studied combinatorial optimization problems. Many researchers (e.g., Grötschel and Wakabayashi, 1989; Ji and Mitchell, 2005; Ji and Mitchell, 2007) have conducted various types of problems both in practical and theoretical fields, including maximum clique or K-equipartition problems with several categories of constraints including minimum or maximum clique size and capacity. The closest work to the proposed problem is the uncapacitated clustering problem (Mehrotra and Trick, 1998), which introduced a B&P algorithm. However, even though the Ryan-Foster branching strategy is incorporated in their work, the solution method is not applicable to the problem in this paper because their objective was to minimize the sum of the arc cost under the minimum clique size requirement. As mentioned earlier, capacity-related constraints (e.g., minimum/maximum clique size) are not considered in this research as we focus instead on the effects of the geometrical coverage constraint. In further research, more practical constraints could be considered with the related knowledge, including cutting plane or branching strategies from the literature mentioned earlier.

The cost of the approximation is the loss of the feasible solution, which might decrease the optimal value of the problem. However, in the case of large-sized problems, the approximation provided time-efficient solutions at relatively high speeds. The performance of the approximated algorithm was analyzed in the perspective of the approximation rate and the computational experiments in Sections 3.4 and 4.

3.4. Comparison of the approximation models

The proposed approximation models are compared from the perspective of the performance ratio. Let $\mathcal{F}$ be the solution of an exact or approximation model in the form of the set of subsets of demand points. For an optimal solution of the USCP $\mathcal{F}^*$, each element $S_j^* \in \mathcal{F}^*$ can be covered by a UAV. Let $\mathcal{F}^{PCA}$ and $\mathcal{F}^{DA}$ be optimal solutions for the PCA model and the DA model, respectively. An approximation ratio of each approximation model is calculated based on the ratio between $|\mathcal{F}|$ and $|\mathcal{F}^*|$.

Theorem 3. *The PCA model is a 3-approximation for the USCP.*

It is obvious that if a set of demand points can be enclosed in a circle with a given radius R, the PCA model requires at most three circles with the same radius to cover the set. The worst-case is when the solution of the PCA model separates the given set into three sets whose diameters are equal to $\sqrt{3}R$. This case is presented in Fig. 2.

Proof. For every element $S_j^* \in \mathcal{F}^*$, there exists a subset $\mathcal{P}^{PCA} \subseteq \mathcal{F}^{PCA}$ such that $|\mathcal{P}^{PCA}| \leq 3$ and $S_j^* \subseteq \bigcup \mathcal{P}^{PCA}$. It will follow that $|\mathcal{F}^*| \geq |\mathcal{F}^{PCA}|/3$. □

In the DA model, the size of the grid affects the computation speed and the approximation ratio–that is, when the grid size decreases, the loss in the approximation follows until the objective value converges to a near-optimal value of USCP. Nevertheless, the computational burden increases after the size of the problem increases. Therefore, it is essential to decide on an appropriate grid size. From the perspective of the approximation ratio, the most natural setting of the DA model with a grid size of R is used for the analysis. Let G_d and $\mathcal{F}_{G_d}^{DA}$ be the grid size and the associated optimal solution of the DA model, respectively. If $G_d \leq \sqrt{2}R$, the DA model can cover the plane with circles with radius R around each lattice point.

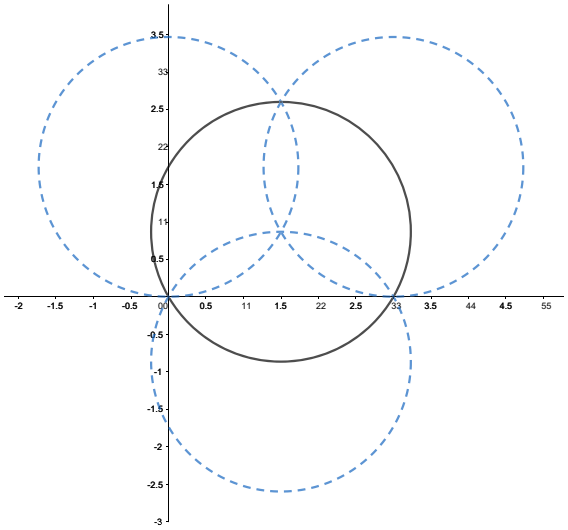


Fig. 2. Worst-case of the PCA.

Theorem 4. *The approximation ratio of DA model with $G_d \geq R$ for the USCP is larger than 3.*

Proof. To prove that this is true, a counterexample that holds $3|\mathcal{F}^*| < |\mathcal{F}_{G_d}^{DA}|$ is suggested: Let $R = \sqrt{3}$ and $|N| = 4$ with $(a_i^x, a_i^y) = (\frac{17}{10}, \frac{\sqrt{11}}{5} - \frac{\sqrt{299}}{10}), (\frac{33}{10}, \frac{3\sqrt{11}}{10}), (\frac{17}{10}, \frac{\sqrt{11}}{5} + \frac{\sqrt{299}}{10})$, and $(-\frac{1}{10}, \frac{3\sqrt{11}}{10})$ are given. The USCP covers the given demand points with one circle: $(x - \frac{16}{10})^2 + (y - \frac{\sqrt{11}}{5})^2 = 3$. Therefore, $|\mathcal{F}^*| = 1$ holds. In the DA model with $G_d = \sqrt{3}$, no lattice point that can cover more than one given demand point. Thus, $|\mathcal{F}_R^{DA}| > 3$ holds for the given counterexample. □

Note that the approximation ratio is measured based on the worst-case scenario. In most situations, the worst cases have the extreme position of the demand points that spreading around circles with the coverage radius. In most instances, the objective value of the approximated model was within the 30% gap from the optimal value of USCP. Section 4 compares performances of the DA and the PCA models with the exact model based on the computation speed and the objective value.

3.5. Framework of the solution algorithm for the PCBP model

Fig. 3 shows the overall framework of the solution algorithm using the PCBP model.

The framework consists of three phases: approximation, B&P, and flight position decision phase. At the first phase, information on the positions of demand points is translated into a set of pairwise-conflict constraints based on Jung's theorem. Constraint (21) is calculated for each pair of demand points i and k based on the distance d_{ik} and M. At the second phase, the PCBP algorithm is executed for the set covering solution; this algorithm provides only the set of demand points assigned for each UAV, unlike the EBP algorithm, which also determines the position for each UAV. Therefore, an additional phase is required to decide the position

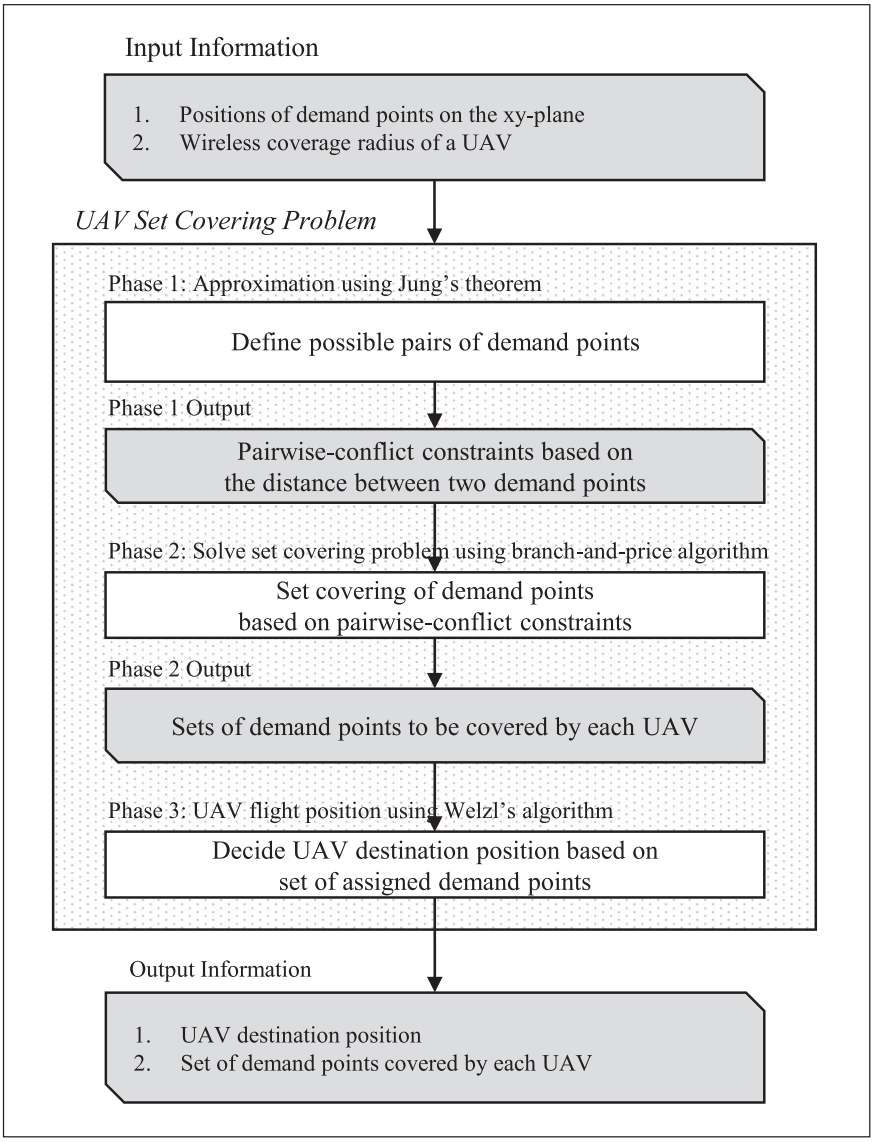


Fig. 3. Framework of the solution algorithm for PCBP model.

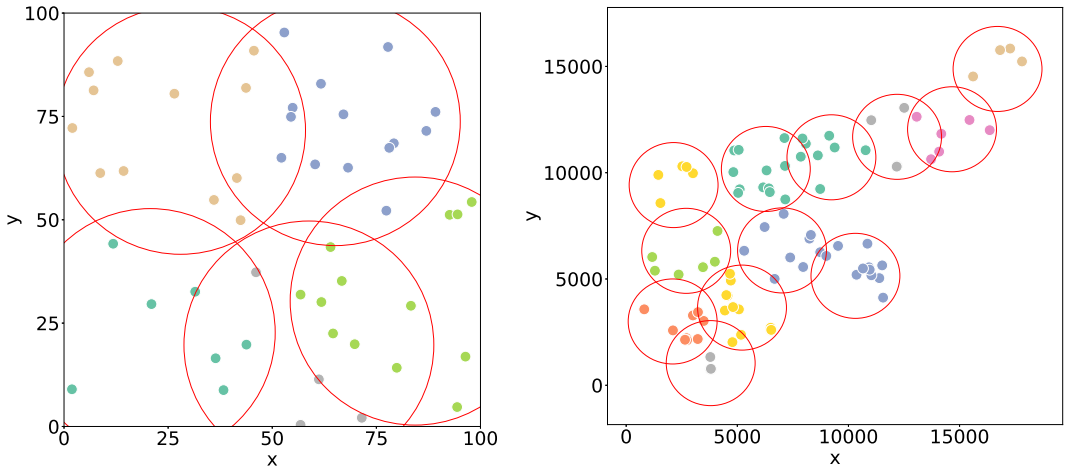
for each UAV. In the third phase, Welzl's algorithm is employed for each UAV to provide the destination points. Since Welzl's algorithm provides not only the circumcenter but also the circumradius of the given set, the feasible region that the UAV needs to fly to cover the demand points can be calculated based on the output solution. Figs. 4a and 4b show solutions of the USCP, illustrated on the xy-plane.

4. Computational experiments

We conducted computational experiments to measure the performance of the proposed solution algorithms (Section 4.2). Section 4.1 describes two datasets used for computational experiments. The Euclidean standard (ES), Euclidean branch-and-price (EBP), pairwise-conflict constraint approximated standard formulation (PCS) and pairwise-conflict constraint approximated branch-and-price algorithm (PCBP) models were developed in FICO Xpress 7.9 and solved with Xpress-Optimizer 33.01.02. MIQCP in the ES and EBP models were solved by B&B in Xpress MIQCQP solver using barrier algorithm for each nodes. The discrete approximation (DA) model was developed in FICO Xpress Python interface 8.6.1. and solved on Python 3.6. Experiments were performed with Intel ® Core ™ i7-3820 CPU at 3.60 GHz and 24 GB of RAM operated on a Windows 10 64-bit operating system.

4.1. Datasets used in the experiments

Two datasets were used for the computational experiments. A small-sized artificial dataset drew input from the benchmark data of the customer position of a capacitated p-median test problem in OR-Library (Beasley, 1990), which was introduced by Osman and Christofides (1994). Based on the benchmark data, instances were developed with three sizes of demand points: 10, 20, and 50. For each size of the demand point, 10 instances were modeled by distributing demand points uniformly on the 100 × 100 xy-plane. Three coverage radii-10, 20, and 30-were tested for each instance. A realistic-scale dataset was developed based on the well-known dataset of Hurricane Katrina Fatalities (HKF), as reported by Maaskant et al. (2018). HKF dataset includes data on the recovery of deceased victims of Hurricane Katrina, one of the most notorious hurricanes ever faced by the United States. The HKF dataset was used to measure the applicability of the proposed algorithm in the actual situation and was named as a realistic-scale dataset in this research. Detailed information on fatalities in the



(a) Artificial dataset, $|N|$=50, Rd=30 (b) Realistic-scale dataset, $|N|$=100, Rd=2000

Fig. 4. Solution examples.

HKF dataset has been used in a variety of studies on disaster management (Brunkard et al., 2008; Jonkman et al., 2009). The recovery data consists of GPS coordinates, type of recovery location, and dates of the recovery of 771 fatalities of Louisiana. For the realistic-scale dataset, 20 instances and 60 problems were generated based on the 539 fatalities found in New Orleans. Fatalities were considered as demand points if the person could have survived if they had wireless communication. Datasets using two sizes of demand points-50 and 100-were developed. For each size of demand points, 10 instances were generated by randomly picking demand points from the original dataset. The GPS information was translated into Cartesian coordinates by equirectangular projection.

Using state-of-the-art wireless communication technology, UAVs can construct a network with distances from 200 m to 1,000 m (Chand et al., 2018; Gu et al., 2015). Thus, three radii-200, 1,000, and 2,000 m-were tested for each instance. Fig. 4b shows a solution of a realistic-scale dataset. In total, there were 15 classes of problems, each consisting of 10 instances; 150 different problems ranging from small to large were conducted for the computational experiments.

4.2. Algorithmic performances

We compared the performances of the ES, DA, EBP, PCS, and PCBP. For each experiment, the limitation of maximum computation time was set to 3000 seconds because rapid computation is extremely important in disaster management. We conducted three analyses for the algorithmic performances. First, we summarized the computational experiments and compared the algorithmic performance between the proposed algorithms from the perspectives of computation time and optimality. A detailed analysis on the DA model is presented, including the adequate grid size and the comparison between the DA and the PCBP model. Second, we conducted further analysis of the B&P algorithm. Observations of root node CG and the overall B&P algorithms were executed for the EBP and the PCBP. The comparison consisted of the quality of the root node LP bound, the number of columns generated for each stage, and the computation times. Third, we provided a sensitivity analysis of the proposed algorithms for managerial insight.

For the first analysis,Tables 1–3 summarize the computational results of five algorithms, which are related to the computation speed and optimality, respectively. The columns in these tables are defined as follows:

- $|N|$: the number of demand points
- Rd: the coverage radius of a UAV
- #*Opt*: the number of problems for which the algorithm provided the optimal solution
- #*Feas*: the number of problems for which the algorithm provided at least one feasible solution
- *Time*: the average time for the computation to find the optimal solution– For problems not solved within the time limit, the limit was used as the computation time while calculating the average.
- Gap_L: the average of the gap between the best lower bound (BB) and the best feasible solution (BFS)–For the problems for which the algorithm provided optimal solutions, $Gap_L = 0$ because the BB meets the BFS at the optimal solution. Gap_L was used to evaluate the convergence of an algorithm itself.

$$Gap_L = \frac{(\text{BFS}) - (\text{BB})}{(\text{BFS})} \times 100\%$$

- # of UAVs: the average of the BFS, or the number of UAVs needed to cover every demand point
- *Gap*: the average of the gap between # of UAVs of the EBP and an algorithm–*Gap* was used to assess the optimality of an algorithm comparing it with the EBP. It was possible for a problem and an algorithm to have either positive or negative *Gap*.

$$Gap = \frac{(\text{BFS of an algorithm}) - (\text{BFS of EBP})}{(\text{BFS of EBP})} \times 100\%$$

In the field of aerial operation, some circumstances demand sub-minute or sub-second time pressures. Table 1 shows that the EBP could solve almost every problem within 100 seconds, and the PCBP could solve every problem class except $|N| = 100$ and Rd =2,000, faster than the EBP. The computation speed between the EBP and the PCBP depends on $|N|$ and Rd, both of which affect the sparsity of the problem. In the USCP, the *sparsity* of the problem is affected not only by the density of demand points but also by the coverage radius of the UAVs. In *sparse* problems, demand points are widely spread with a small coverage radius, resulting in a relatively small number of demand points assigned to a UAV. There is a tendency for the computation speed of the PCBP to be faster than that of the EBP when the problem class is sparse. In the artificial dataset, when Rd was less than 30, the PCBP was faster than the EBP for 54 out of 60 instances. However, when Rd was 30,

Table 1
Results related to the computation speed.

$\|N\|$	Rd	$\#Opt/\#Feas$				Time(s)				$Gap_L(\%)$			
		ES	EBP	PCS	PCBP	ES	EBP	PCS	PCBP	ES	EBP	PCS	PCBP
10	10	2/10	10/10	10/10	10/10	2862.10	0.29	0.01	0.01	27.82	0.00	0.00	0.00
	20	10/10	10/10	10/10	10/10	91.26	0.32	0.01	0.01	0.00	0.00	0.00	0.00
	30	10/10	10/10	10/10	10/10	8.55	0.25	0.01	0.01	0.00	0.00	0.00	0.00
20	10	0/2	10/10	10/10	10/10	3000s	0.99	0.04	0.03	82.06	0.00	0.00	0.00
	20	0/10	10/10	10/10	10/10	3000s	1.09	0.09	0.06	45.00	0.00	0.00	0.00
	30	5/10	10/10	10/10	10/10	1531.02	0.68	0.04	0.20	12.50	0.00	0.00	0.00
50	10	0/0	10/10	9/10	10/10	3000s	10.39	393.99	2.02	96.00	0.00	0.77	0.00
	20	0/0	10/10	8/10	10/10	3000s	14.46	663.61	22.49	96.00	0.00	2.27	0.00
	30	0/8	10/10	10/10	10/10	3000s	8.35	3.15	70.02	68.05	0.00	0.00	0.00
50	200	0/0	10/10	10/10	10/10	3000s	3.10	0.79	0.18	95.80	0.00	0.00	0.00
	1000	0/0	10/10	10/10	10/10	3000s	6.69	110.15	1.08	95.60	0.00	0.00	0.00
	2000	0/0	10/10	10/10	10/10	3000s	7.07	95.14	7.38	95.75	0.00	0.00	0.00
100	200	0/0	10/10	10/10	10/10	3000s	32.02	360.90	3.72	98.98	0.00	0.00	0.00
	1000	0/0	10/10	0/10	10/10	3000s	87.32	3000s	74.37	98.82	0.00	12.31	0.00
	2000	0/0	10/10	1/10	10/10	3000s	69.65	2704.66	771.47	98.98	0.00	11.19	0.00

3000s: The solver failed to find the optimal solution within 3000s for every instance.

Table 2
Results related to the optimality.

$\|N\|$	Rd	# of UAVs				$Gap(\%)$		
		ES	EBP	PCS	PCBP	ES	PCS	PCBP
10	10	7.0	7.4	7.9	7.9	0.00	7.26	7.26
	20	3.9	4.1	4.8	4.8	0.00	19.17	19.17
	30	2.8	2.8	2.9	2.9	0.00	5.00	5.00
20	10	20.0	11.1	12.2	12.2	67.20	10.15	10.15
	20	5.4	5.5	6.4	6.4	0.00	17.00	17.00
	30	3.5	3.5	4.0	4.0	0.00	16.67	16.67
50	10	50.0	17.0	19.3	19.3	195.18	13.87	13.87
	20	50.0	7.2	8.6	8.6	593.43	19.64	19.64
	30	14.5	4.0	4.8	4.8	262.50	20.00	20.00
50	200	50.0	39.0	40.2	40.2	28.55	3.06	3.06
	1000	50.0	17.4	19.9	19.9	188.93	14.45	14.45
	2000	50.0	9.1	10.8	10.8	452.78	19.03	19.03
100	200	100.0	65.3	68.0	68.0	53.39	4.08	4.08
	1000	100.0	22.1	25.5	25.2	354.20	15.49	14.14
	2000	100.0	10.0	12.0	11.9	912.37	21.12	19.87

Table 3
Performance of DA in accordance of grid sizes related to Rd.

$\|N\|$	Rd	Time(s)			# of UAVs		
		Grid size of DA			Grid size of DA		
		$\sqrt{2}Rd$	Rd	$Rd/2$	$\sqrt{2}Rd$	Rd	$Rd/2$
10	10	0.050	0.052	0.129	8.9	8.5	7.8
	20	0.011	0.010	0.030	6.9	5.8	4.7
	30	0.007	0.007	0.015	4.4	4.1	3.4
20	10	0.065	0.110	0.403	15.8	13.7	12.3
	20	0.025	0.029	0.095	9.7	8.1	6.3
	30	0.013	0.016	0.046	6.2	5.4	4.3
50	10	0.274	0.478	**1.695**	27.7	23.3	**19.1**
	20	0.071	0.093	0.280	13.7	11.2	8.8
	30	0.038	0.052	0.123	8.4	7.2	5.1
50	200	78.858	300.222	3885.516	41.5	40.8	40.0
	1000	0.860	1.629	9.543	26.4	22.8	19.8
	2000	0.193	0.313	**1.169**	14.7	13.4	**10.8**
100	200	320.876	822.009	10633.638	73.9	70.7	67.3
	1000	3.682	6.733	32.115	35.1	30.5	25.6
	2000	0.660	1.115	**3.787**	18.2	15.1	**11.9**

the EBP was faster than the PCBP for 11 out of 30 instances. Likewise, in the realistic-scale dataset, when Rd was less than 2,000, the PCBP was faster than the EBP for 35 out of 40 instances, while only 6 out of 10 instances were faster when Rd =2,000.

In some extremely dense problem classes in artificial dataset $Rd = 30$, the computation speed of the PCS was faster than both the EBP's and the PCBP's. Fig. 4a shows the PCS solution of Instance 1 in problem class $|N| = 50$ and $Rd = 30$. The 100×100 plane can almost be covered by 5 UAVs with a coverage radius of 30. Thus, the solution does not change much as $|N|$ increases, which means that the concurrent optimizer of the commercial solver provides the solution fast.

Table 4
Performance of DA in accordance of grid sizes.

$\|N\|$	Rd	Time(s)						# of UAVs					
		Grid size of DA						Grid size of DA					
		32	16	8	4	2	1	32	16	8	4	2	1
10	10	–	–	0.046	0.221	1.772	22.888	–	–	7.9	7.7	7.7	7.5
	20	–	**0.013**	0.045	0.213	1.747	22.472	–	4.8	4.5	4.2	4.1	4.1
	30	0.006	0.013	0.043	0.216	1.875	23.778	4.2	3.2	2.9	2.9	2.9	2.8
20	10	–	–	0.135	0.667	5.596	88.567	–	–	12.8	12	11.7	11.5
	20	–	0.034	0.125	0.611	5.341	91.234	–	7	6.2	5.7	5.6	5.6
	30	0.014	0.032	**0.120**	0.613	5.346	93.324	5.2	4.1	**3.7**	3.6	3.5	3.5
50	10	–	–	0.669	2.756	20.481	302.785	–	–	21.2	18.6	17.8	17.4
	20	–	0.137	**0.446**	**2.154**	18.057	308.372	–	9.9	**8.4**	**7.7**	7.5	7.4
	30	0.047	0.113	**0.377**	**1.907**	**17.278**	295.531	7.6	5.5	**4.5**	**4.1**	**4**	4
			2048	1024	512	256	128		2048	1024	512	256	128
50	200		–	–	–	122.536	1681.441		–	–	–	41.5	40.1
	1000		–	1.823	9.508	110.616	1671.042		–	23.7	19.3	18.3	17.9
	2000		0.339	**1.308**	8.007	103.741	1677.347		13.2	**10.8**	9.9	9.5	9.2
100	200		–	–	–	502.397	4295.910		–	–	–	72.3	68.7
	1000		–	7.117	35.634	311.769	3678.380		–	31.3	25.6	23.3	22.5
	2000		1.079	3.906	**24.819**	**280.069**	3714.565		15.4	12.1	**10.8**	**10.4**	10.3

– : Grid size is bigger than the upper limit.

Table 2 shows the loss of optimality of the PCS and the PCBP. There was a general tendency for the gap to improve when the problem class became denser. The problems in the artificial dataset had demand points on the fixed-size xy-plane; the larger number of demand points $|N|$ denoted the denser distribution of demand points. The tendency mentioned above could be seen when comparing the problem classes in the artificial dataset with the same Rd but different $|N|$. In some cases, the approximation of the PCS and the PCBP provided dramatic increases in *Gap*. However, *Gap* measured in the ratio can be exaggerated when the objective value is too small. For example, for an instance with $|N| = 50$ and $Rd = 30$, even though the objective value of the PCBP (which is 5) is only 1 larger than the optimal value (which is 4), the *Gap* equals 25%. In total, for 10 out of 150 instances the actual difference of the objective value between the PCBP and the EBP was greater than 3.

Tables 3 and 4 show the performance of the DA according to the grid size based on two factors. Because the DA experiment was performed on the Python and FICO Xpress Python interface with a large-sized problem, the computation was executed even after the time limit had passed (3,000 seconds). The problem classes in which the DA showed shorter average computation times and better average BFS than the PCBP are indicated in bold font. In the problem classes with $(|N|, Rd) = (50, 20), (50, 30)$, and $(100, 2{,}000)$, the system could be covered perfectly with a number of UAVs equal to about one-tenth the number of demand points. Thus, it is reasonable to assume that for extremely dense problems, discretization can provide a good solution efficiently.

However, it is difficult to find a simple, standardized way to decide the grid size. The most natural way is to base the grid size on the multiples of Rd. As discussed in Section 3.4, the upper limit of grid size G_d is $\sqrt{2}Rd$. Table 3 compares three grid sizes–$\sqrt{2}Rd$, Rd, and $Rd/2$. For $G_d = Rd$, in 98 out of 150 instances, the DA found the solution faster than the PCBP. However, the DA had only 22 instances with the same objective value of the PCBP and no instance with the lower objective value. For only five instances did the DA perform better than the PCBP in terms of both computation speed and optimality. For $G_d = Rd/2$, the DA had only 61 instances with shorter computation times than the PCBP; however, the DA also had 22 instances with the better objective value and 76 instances with the same objective value. In conclusion, the DA with a grid size of Rd was faster than the PCBP but was not sufficient in terms of optimality. At the same time, the DA with the grid size of $Rd/2$ was better than the PCBP in terms of optimality, though it was also slower.

Instead of finding one standardized grid size as the multiple of Rd, problem classes in which the DA performed well were noticed. For problems with more density, the DA outperformed the PCBP and could find near-optimal solutions within a shorter amount of time than the EBP. Table 4 shows that for dense problems, the DA with a grid size smaller than $Rd/4$ could be solved faster than both the PCBP and the EBP. Appendix A compares the computation times and the objective values of the EBP, PCBP, and DA with various grid sizes.

For the second analysis, we summarized the performance of the B&P algorithm. The computation performances of the root node and the overall B&P algorithm are listed separately. Table 5 compares the integrality gap, the number of columns generated during the algorithms, and the computation time of the root nodes and their ratio compared to the overall B&P algorithm. The reformulation of the problem and the CG in the root node provided strong LP bound, which minimized the branching while solving the problem. The LP bounds and the process of branching are illustrated in Table 6. The columns in Tables 5 and 6 are defined as follows:

- Integrality Gap: the average of the ratio of the BFS over the lower bound-For the root node, the LP solution of the root node was used for the lower bound. We used different BFS to the EBP and the PCBP, because the solution space of the EBP included the solution space of the PCBP. For the EBP, some problems could not be solved within the time limits. The better feasible solutions found during the four algorithms were used as the BFS for those unsolved problems. In addition, for the PCBP, there was one unsolved problem, and the BFS of the PCS replaced the PCBP's.

$$\text{Integrality Gap} = \frac{(\text{BFS})}{(\text{Lower bound})}$$

- # Columns: the average of the number of columns generated while solving the root node and the overall B&P algorithm
- Root Time: the average computation times needed to solve the root node
- Time Ratio: the ratio of the computation times of the root node over the overall B&P algorithm

$$\text{Time Ratio} = \frac{(\text{Root Time})}{(Time)} \times 100\%$$

- #IB: the number of problems that were solved for which the LP bound was not changed by branching

Table 5
Comparison between branch-and-price algorithms.

|N|	Rd	Integrality gap				# Columns				Root time(s)		Time ratio(%)	
		EBP		PCBP		EBP		PCBP		EBP	PCBP	EBP	PCBP
		Root	B&P	Root	B&P	Root	B&P	Root	B&P				
10	10	1.009	1.000	1.000	1.000	3.8	3.9	3.0	3.0	0.20	0.00	66.02	63.25
	20	1.032	1.000	1.000	1.000	6.0	7.0	6.3	6.4	0.23	0.01	68.73	76.98
	30	1.000	1.000	1.000	1.000	5.3	5.3	7.7	7.7	0.20	0.01	79.65	76.19
20	10	1.005	1.000	1.000	1.000	9.9	10.0	8.2	8.3	0.90	0.03	90.18	80.48
	20	1.009	1.000	1.000	1.000	10.2	10.6	13.9	14.2	0.90	0.05	84.49	84.02
	30	1.000	1.000	1.000	1.000	8.0	8.5	17.7	22.1	0.58	0.12	87.90	80.69
50	10	1.006	1.000	1.000	1.000	25.2	25.9	36.0	39.2	9.65	1.58	93.39	82.81
	20	1.016	1.000	1.018	1.000	27.6	32.9	100.5	117.5	11.86	15.44	86.81	78.57
	30	1.000	1.000	1.000	1.000	16.6	18.7	184.6	211.5	6.80	53.26	90.95	80.98
50	200	1.000	1.000	1.000	1.000	9.5	9.5	8.4	8.4	2.75	0.14	88.29	75.34
	1000	1.006	1.000	1.003	1.000	19.2	19.8	24.9	27.2	6.14	0.82	92.05	79.89
	2000	1.018	1.000	1.005	1.000	18.9	20.1	57.8	71.4	6.46	4.17	91.29	71.23
100	200	1.000	1.000	1.000	1.000	21.4	21.4	20.7	20.7	30.30	3.34	94.53	89.67
	1000	1.010	1.000	1.000	1.000	38.9	41.6	86.5	99.8	80.33	51.48	92.91	82.83
	2000	1.010	1.000	1.004	1.000	29.6	32.8	243.5	310.6	63.21	454.32	91.24	64.70

Table 6
The change of the LP bound over the branching.

|N|	*Rd*	EBP		PCBP		Nodes	
		#IB	#B	#IB	#B	EBP	PCBP
10	10	0	1	0	0	1.1	1.0
	20	1	3	1	1	1.6	1.1
	30	0	0	0	0	1.0	1.0
20	10	0	1	1	1	1.1	1.1
	20	0	1	1	1	1.2	1.2
	30	1	1	2	2	1.2	1.6
50	10	1	3	5	5	1.4	2.2
	20	0	3	3	6	2.5	4.1
	30	1	1	4	4	1.2	2.9
50	200	0	0	0	0	1.0	1.0
	1000	0	2	4	5	1.3	1.8
	2000	2	5	6	7	1.6	3.1
100	200	0	0	0	0	1.0	1.0
	1000	2	6	6	6	2.2	2.6
	2000	3	5	9	10	2.1	4.3

Table 7
The number of demand points assigned to one UAV.

Rd	|N|	EBP		PCBP	
		Avg.	Max.	Avg.	Max.
10	10	1.4	2.4	1.3	2.3
	20	1.8	3.3	1.7	3.1
	50	3.0	5.9	2.6	5.3
20	10	2.5	4.0	2.1	3.6
	20	3.7	6.3	3.1	5.5
	50	7.0	11.7	5.8	10.0
30	10	3.8	5.3	3.6	4.7
	20	5.8	8.7	5.0	7.7
	50	12.5	18.8	10.5	14.9
200	50	1.3	3.7	1.2	3.6
	100	1.5	7.2	1.5	7.2
1000	50	2.9	7.5	2.5	6.8
	100	4.5	13.9	4.0	11.5
2000	50	5.5	11.8	4.7	9.5
	100	10.1	23.9	8.4	19.3

- #B: the number of problems that were solved for which branching was executed
- Nodes: the average number of nodes in the branch-and-bound tree

As mentioned in Section 3.1, the LP relaxation neutralized the coverage constraints (4) and (20) in the ES and the PCS, respectively. In the extended formulation, however, even the LP solutions satisfied the coverage constraints and had stronger LP bounds. Accordingly, unlike most of the literature on the B&P approach, it was not significant to compare the LP bound between standard formulations and the root node LP bounds of B&P algorithms; instead, we compared the Integrality Gap between two B&P algorithms, as shown in Table 5. Both EBP and PCBP had strong root node LP bounds, considering that both algorithms had an integrality gap of root node near 1 for almost every problem class. Notably, many problems had an integrality gap of root node equal to 1, which meant that the value of the root node LP bound was the same as for the BFS. Likewise, the number of columns generated and the computation times were highly concentrated on root nodes. Table 6 shows the change of the LP bound over branching for a detailed analysis of the performances of the CG algorithm on the root node. The EBP and the PCBP provided a small integrality gap as the nominal set covering problem and had a minimal number of branches. Furthermore, after the root node CG was finished, a limited number of problems (32 of 150 problems in the EBP and 48 of 150 problems in the PCBP) had branching. Even if the branching occurred, many problems did not have any improvement of LP bounds (11 of 32 problems in the EBP and 32 of 48 problems in the PCBP) until the algorithm found its optimal solution. For dense problem classes, as the computation times of PCBP increases, the number of the generated columns followed. It is analogized that in the dense problems, the possible combination of the demand points satisfying Constraint (21) increased quickly and slacked the computation speed.

For the third analysis, we provided insights for the decision makers of disaster management. In the realistic-scale dataset, the size of the xy-plane was not limited, so the sparsity of the problem was more related to the |*N*| than *Rd*. In the real-world, we recommend the authorities to use the PCBP in the sparse disaster situations, sparsely distributed survivors or UAVs of smaller radius. For the dense situations, the EBP or the DA with a small grid size is recommended. Table 7 lists the number of demand points assigned to one UAV (#DM), which is directly related to the sparsity of the problem. At the same time, it can be used to predict the scale of the network traffic and to estimate capacity. In a realistic-scale dataset, even though the average number of demand points covered by one UAV maintained a reasonable size, the maximum #DM exceeded the realistic limitation of the traffic. However, considering that the distribution of #DM was skewed to the

left, the solution would not change dramatically even if we were to consider an additional constraint of an upper bound of #DM. We could further speculate that in a real disaster, survivors are distributed sparsely enough for the network transmission capacity to be immaterial; in that case, the solution of the USCP could be used without the capacity constraint. The sparsity of demand points also hinders the improvement effect of network coverage. The increase of *Rd* did not bring the same amount of increase of #DM. Even though *Rd* increased fivefold, from 200 m to 1,000 m, #DM increased only around twofold. On the other hand, for increased $|N|$, #DM also grew, even though *Rd* did not change. Thus, the number of UAVs required increased, but the growth rate was less than 1. The results in Tables 2 and 7 can be used to plan the cost-effective development objective of UAVs and wireless network router. Based on the specifics of the developed system, the decision maker can scale the required number of UAVs and make a response plan.

5. Conclusions

This paper introduced the problem of developing a flight plan for UAVs to provide a wireless network in the shadowed area of a disaster environment. We defined the USCP as a set covering problem with a fixed coverage radius constraint in Euclidean distance and without predefined candidates of positions. Due to the quadratic constraints, a standard formulation of the USCP was not solvable even for the smallest problems. A simple discrete approximation model is proposed, and the approximation ratio of the DA model with the grid size equal to the coverage radius is analyzed. An extended formulation and the associated B&P algorithm, which were developed for the stronger LP bound, showed faster computation speed. Based on the Ryan-Foster branching strategy used for the B&P algorithm, we implemented Jung's theorem to approximate the quadratic coverage constraint of the USCP into the linear pairwise-conflict constraint. The approximation decomposed the decisions of the USCP into two separate decisions-UAV position and set partition-and made only the set partition decision relevant. The computational results showed that the EBP and the PCBP were applicable for both small-sized artificial and realistic-scale problems within a proper time limit. For sparse problems, the PCBP provided the near-optimal solution faster than the EBP, and for dense problems, the DA could find a better solution faster than the PCBP.

For future research, a heuristic algorithm to develop a powerful initial column is expected to increase the computation speed, considering the effective LP bounds provided by B&P algorithms. To use the full capacity of UAVs, practical restrictions and possible extensions should be applied in the USCP. The overlap interference among UAVs, transmission capacity, and shadowing effects by obstacles should be considered when creating the flight plan. There is also an additional freedom of the flight altitude, which will change the network coverage radius and the aforementioned effects. Further research in different directions is also required, to address uncertainty or incomplete information on demand points because of the importance of the robust solution in disaster management.

CRediT authorship contribution statement

Youngsoo Park: Conceptualization, Methodology, Software, Investigation, Data curation, Writing - original draft, Writing - review & editing, Visualization. **Peter Nielsen:** Supervision. **Ilkyeong Moon:** Conceptualization, Validation, Writing - review & editing, Supervision.

Acknowledgements

The authors are grateful for the valuable comments from the area editor and four anonymous reviewers. This research was supported by the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT & Future Planning [Grant number NRF-2019R1A2C2084616].

Appendix A. Comparison of the computation times and objective value of the proposed algorithms

Figs. A.5–A.8 illustrate the effects of the DA model's grid size on the computation times and objective value. In each figure, either nine or six problem classes are arranged in matrices, showing the performance of the DA model over the different grid sizes; that are juxtaposed with the outputs of the EBP and the PCBP. The computation times increase exponentially when the grid size decreases. As is shown, it is difficult to find a universal value or a standardized way to decide an appropriate grid size. However, for dense problems, the DA model with a grid size of one-fourth of the coverage radius could find near-optimal solutions within a reasonable time.

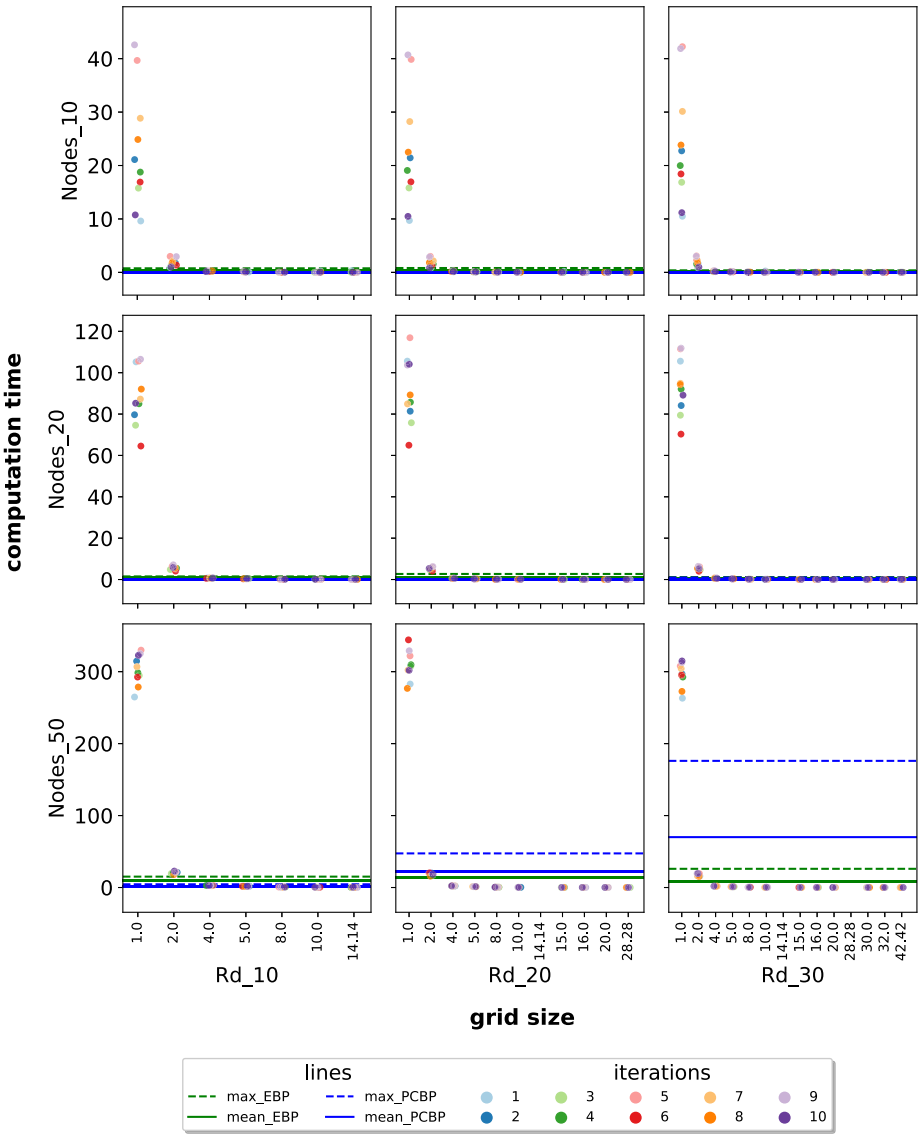


Fig. A.5. Computation time of small-sized artificial problems.

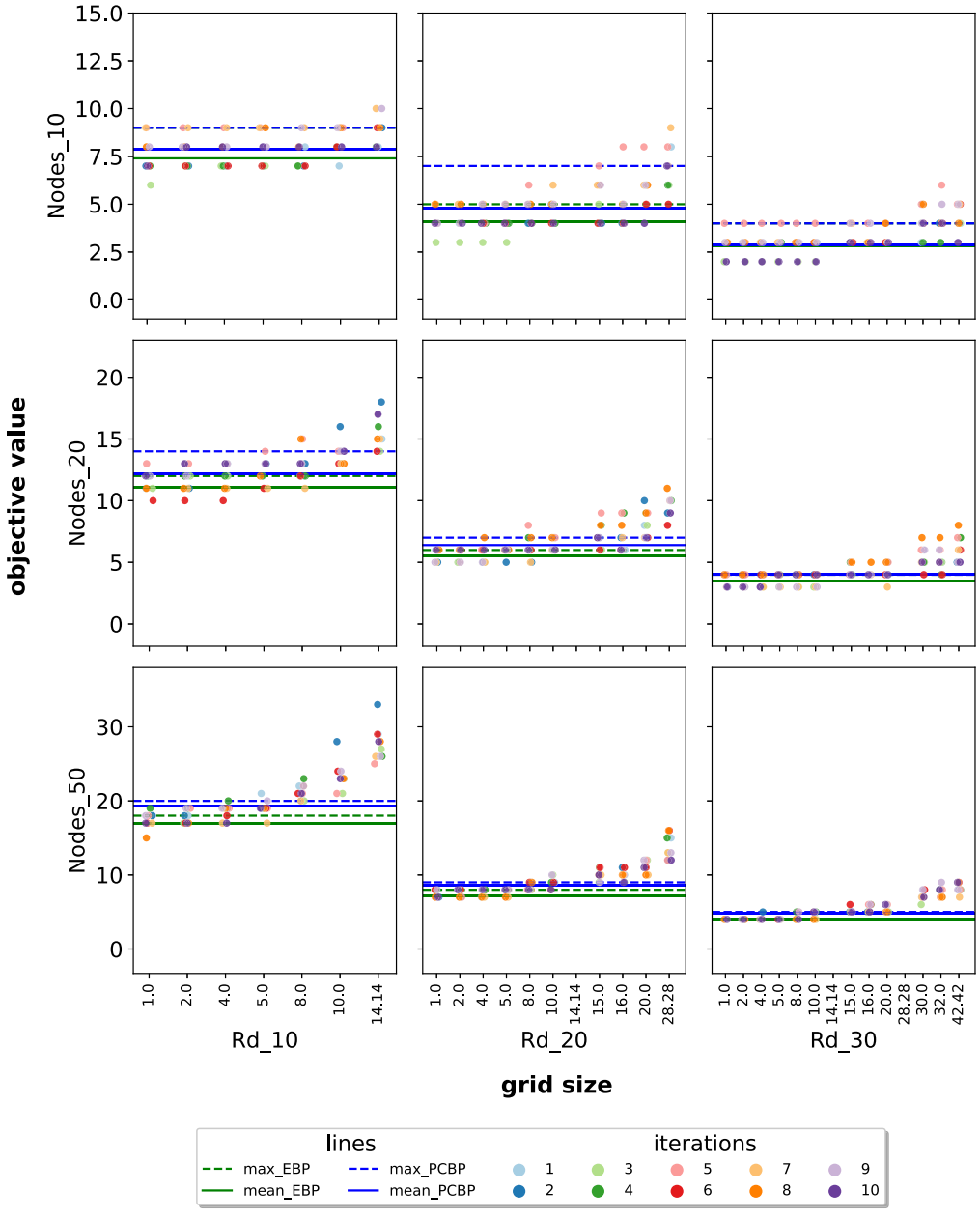


Fig. A.6. Objective value of small-sized artificial problems.

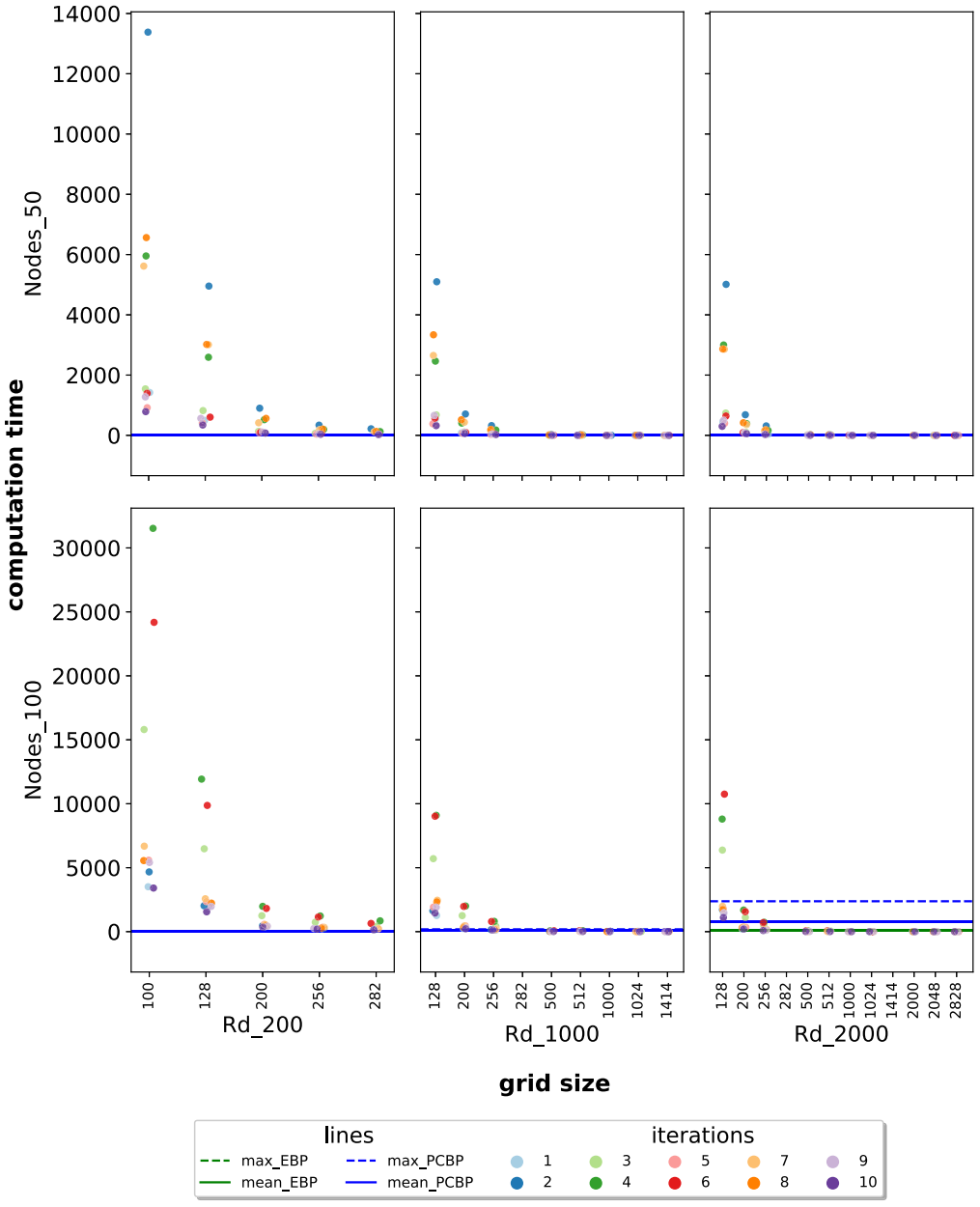


Fig. A.7. Computation time of realistic-scale problem.

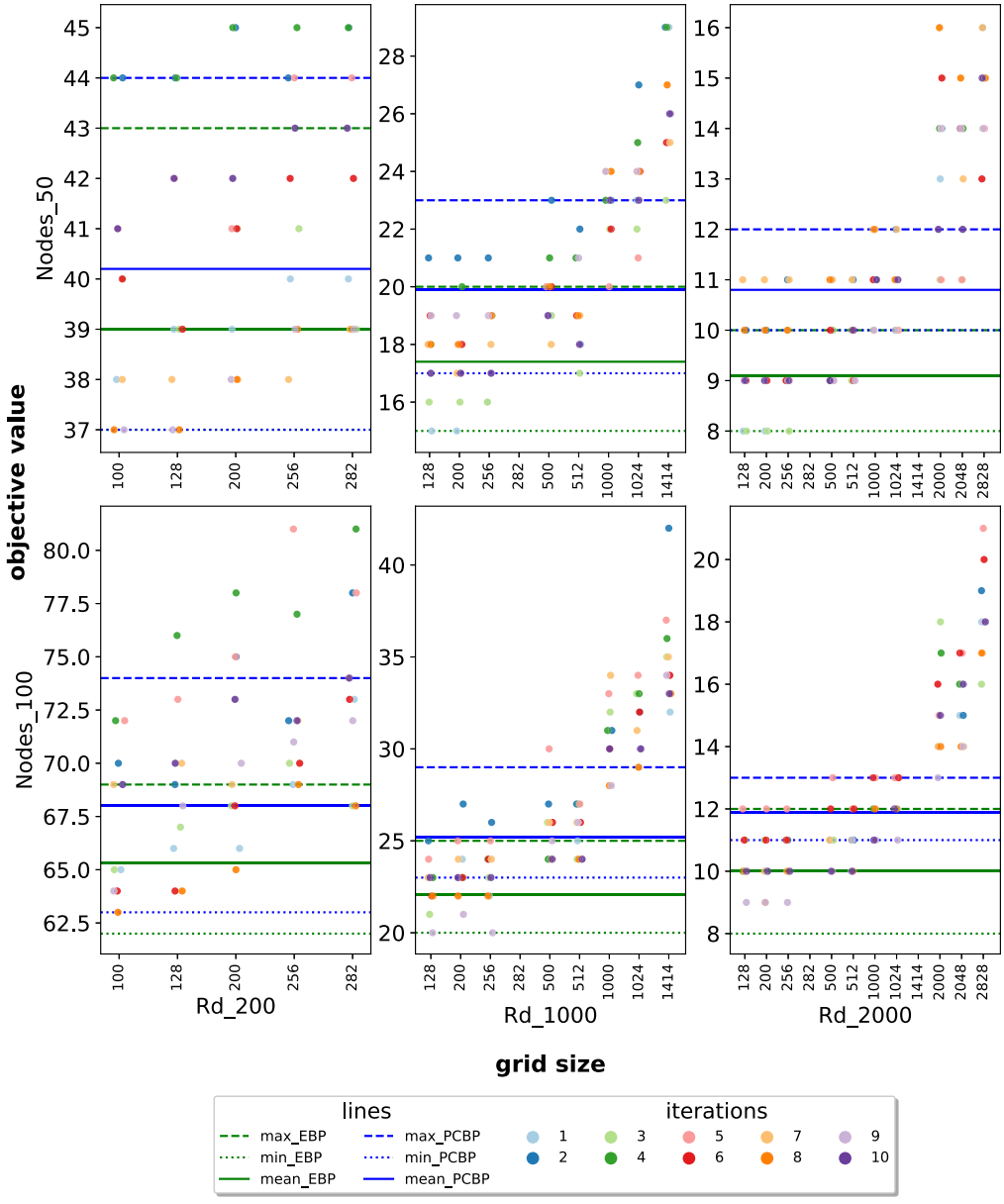


Fig. A.8. Objective value of realistic-scale problem.

References

Ahmadi-Javid, A., Seyedi, P., Syam, S.S., 2017. A survey of healthcare facility location. Comput. Oper. Res. 79, 223–263. doi:10.1016/J.COR.2016.05.018.

Aida, S., Shindo, Y., Utiyama, M., 2013. Rescue activity for the great east japan earthquake based on a website that extracts rescue requests from the net. In: Proceedings of the Workshop on Language Processing and Crisis Information 2013. Asian Federation of Natural Language Processing, pp. 19–25.

Aringhieri, R., Bruni, M., Khodaparasti, S., van Essen, J., 2017. Emergency medical services and beyond: addressing new challenges through a wide literature review. Comput. Oper. Res. 78, 349–368. doi:10.1016/J.COR.2016.09.016.

Beasley, J.E., 1990. OR-library: distributing test problems by electronic mail. J. Oper. Res. Soc. 41 (11), 1069–1072. doi:10.1057/jors.1990.166.

Bélanger, V., Ruiz, A., Soriano, P., 2019. Recent optimization models and trends in location, relocation, and dispatching of emergency medical vehicles. Eur. J. Oper. Res. 272 (1), 1–23. doi:10.1016/J.EJOR.2018.02.055.

Boonmee, C., Arimura, M., Asada, T., 2017. Facility location optimization model for emergency humanitarian logistics. Int. J. Disaster Risk Reduct. 24, 485–498.

Borndörfer, R., Weismantel, R., 2000. Set packing relaxations of some integer programs. Math. Program. 88 (3), 425–450. doi:10.1007/PL00011381.

Brotcorne, L., Laporte, G., Semet, F., 2003. Ambulance location and relocation models. Eur. J. Oper. Res. 147 (3), 451–463. doi:10.1016/S0377-2217(02)00364-8.

Brunkard, J., Namulanda, G., Ratard, R., 2008. Hurricane Katrina deaths, Louisiana, 2005. Disaster Med. Public Health Prep. 2 (4), 215–223. doi:10.1097/DMP.0b013e31818aaf55.

Calik, H., Labbé, M., Yaman, H., 2015. p-center problems. In: Laporte, G., Nickel, S., Saldanha da Gama, F. (Eds.), Location Science. Springer, pp. 79–92. doi:10.1007/978-3-319-13111-5_4.

Capoyleas, V., Rote, G., Woeginger, G., 1991. Geometric clusterings. J. Algorithms 12 (2), 341–356. doi:10.1016/0196-6774(91)90007-L.

Chand, G.S.L.K., Lee, M., Shin, S.Y., 2018. Drone based wireless mesh network for disaster/military environment. J. Comput. Commun. 6 (4), 44–52. doi:10.4236/jcc.2018.64004.

Chandrashekar, K., Dekhordi, M.R., Baras, J.S., 2004. Providing full connectivity in large ad-hoc networks by dynamic placement of aerial platforms. In: IEEE MILCOM 2004. Military Communications Conference, 2004.. IEEE, pp. 1429–1436.

Chowdhury, S., Emelogu, A., Marufuzzaman, M., Nurre, S.G., Bian, L., 2017. Drones for disaster response and relief operations: a continuous approximation model. Int. J. Prod. Econ. 188, 167–184. doi:10.1016/j.ijpe.2017.03.024.

Comley, W.J., 1995. The location of ambivalent facilities: use of a quadratic zero-one programming algorithm. Appl. Math. Model. 19 (1), 26–29.

Daskin, M.S., Maass, K.L., 2015. The p-median problem. In: Laporte, G., Nickel, S., Saldanha da Gama, F. (Eds.), Location Science. Springer, pp. 21–45. doi:10.1007/978-3-319-13111-5_2.

Desaulniers, G., Desrosiers, J., Solomon, M.M., 2006. Column Generation. Springer.

Drezner, Z., Klamroth, K., Schöbel, A., O. Wesolowsky, G., 2001. The weber problem. In: Drezner, Z., Hamacher, H.W. (Eds.), Facility Location: Applications and Theory. Springer, pp. 1–36. doi:10.1007/978-3-642-56082-8_1.

Elzinga, J., Hearn, D., Randolph, W.D., 1976. Minimax multifacility location with euclidean distances. Transp. Sci. 10 (4), 321–336.

Gendreau, M., Manerba, D., Mansini, R., 2016. The multi-vehicle traveling purchaser problem with pairwise incompatibility constraints and unitary demands: a branch-and-price approach. Eur. J. Oper. Res. 248 (1), 59–71. doi:10.1016/j.ejor.2015.06.073.

Grötschel, M., Wakabayashi, Y., 1989. A cutting plane algorithm for a clustering problem. Math. Program. 45 (1), 59–96. doi:10.1007/BF01589097.

Gu, Y., Zhou, M., Fu, S., Wan, Y., 2015. Airborne WiFi networks through directional antennae: an experimental study. In: 2015 IEEE Wireless Communications and Networking Conference (WCNC). IEEE, pp. 1314–1319. doi:10.1109/WCNC.2015.7127659.

Heinzelman, J., Waters, C., 2010. Crowdsourcing crisis information in disaster-affected Haiti. Technical Report. United States Institute of Peace.

Ho, D.-T., Grøtli, E.I., Sujit, P.B., Johansen, T.A., Sousa, J.B., 2015. Optimization of wireless sensor network and UAV data acquisition. J. Intell. Robot. Syst. 78 (1), 159–179.

Hoffman, K.L., Padberg, M., 1993. Solving airline crew scheduling problems by branch-and-cut. Manag. Sci. 39 (6), 657–682. doi:10.1287/mnsc.39.6.657.

Holley, P., 2017. Water is swallowing us up: In Houston, desperate flood victims turn to social media for survival, The Washington Post. (accessed 9 September 2019). https://wapo.st/2vw1F4X?.

Ji, X., Mitchell, J.E., 2005. Finding optimal realignments in sports leagues using a branch-and-cut-and-price approach. Int. J. Oper. Res. 1 (1–2), 101–122.

Ji, X., Mitchell, J.E., 2007. Branch-and-price-and-cut on the clique partitioning problem with minimum clique size requirement. Discrete Optim. 4 (1), 87–102. doi:10.1016/J.DISOPT.2006.10.009.

Johnson, E.L., Mehrotra, A., Nemhauser, G.L., 1993. Min-cut clustering. Math. Program. 62 (1), 133–151. doi:10.1007/BF01585164.

Jonkman, S.N., Maaskant, B., Boyd, E., Levitan, M.L., 2009. Loss of life caused by the flooding of new orleans after hurricane Katrina: analysis of the relationship between flood characteristics and mortality. Risk Anal. 29 (5), 676–698. doi:10.1111/j.1539-6924.2008.01190.x.

Jung, H., 1901. Ueber die kleinste Kugel, die eine räumliche Figur einschliesst.. J. Reine Angew. Math. 123, 241–257.

Kim, D., Lee, K., Moon, I., 2018. Stochastic facility location model for drones considering uncertain flight distance. Ann. Oper. Res. 1–20. doi:10.1007/s10479-018-3114-6.

Kim, S., Moon, I., 2019. Traveling salesman problem with a drone station. IEEE Trans. Syst. Man. Cybern. 49 (1), 42–52. doi:10.1109/TSMC.2018.2867496.

Maaskant, B., Jonkman, S.N., Boyd, E., 2018. Fatalities due to hurricane Katrina (2005). Technical Report. TU Delft doi:10.4121/UUID:CC5A95BB-69AF-4174-80C2-61C69E6109AF.

Manerba, D., Mansini, R., 2016. The nurse routing problem with workload constraints and incompatible services. IFAC-PapersOnLine 49 (12), 1192–1197. doi:10.1016/j.ifacol.2016.07.670.

Mehrotra, A., Trick, M.A., 1998. Cliques and clustering: a combinatorial approach. Oper. Res. Lett. 22 (1), 1–12.

Mozaffari, M., Saad, W., Bennis, M., Debbah, M., 2016. Efficient deployment of multiple unmanned aerial vehicles for optimal wireless coverage. IEEE Commun. Lett. 20 (8), 1647–1650.

Mozaffari, M., Saad, W., Bennis, M., Debbah, M., 2016. Unmanned aerial vehicle with underlaid device-to-device communications: performance and tradeoffs. IEEE Trans. Wirel. Commun. 15 (6), 3949–3963.

Osman, I.H., Christofides, N., 1994. Capacitated clustering problems by hybrid simulated annealing and tabu search. Int. Trans. Oper. Res. 1 (3), 317–336. doi:10.1016/0969-6016(94)90032-9.

Periyasamy, S., Khara, S., Thangavelu, S., 2016. Balanced cluster head selection based on modified k-means in a distributed wireless sensor network. Int. J. Distrib. Sens. Netw. 12 (3), 1–11. doi:10.1155/2016/5040475.

Plastria, F., 2001. Continuous covering location problems. In: Drezner, Z., Hamacher, H.W. (Eds.), Facility Location: Applications and Theory, pp. 37–79. doi:10.1007/978-3-642-56082-8_2.

Ryan, D.M., Foster, B.A., 1981. An integer programming approach to scheduling. In: Wren, A. (Ed.), Computer Scheduling of Public Transport: Urban Passenger Vehicle and Crew Scheduling. North-Holland, pp. 269–280.

Sadykov, R., Vanderbeck, F., 2012. Bin packing with conflicts: a generic branch-and-Pprice algorithm. INFORMS J. Comput. 25 (2), 244–255. doi:10.1287/ijoc.1120.0499.

Sasikumar, P., Khara, S., 2012. K-means clustering in wireless sensor networks. In: 2012 Fourth International Conference on Computational Intelligence and Communication Networks. IEEE, pp. 140–144.

Simchi-Levi, D., Chen, X., Bramel, J., 2005. The Logic of Logistics: Theory, Algorithms, and Applications for Logistics and Supply Chain Management. Springer.

Toregas, C., Swain, R., ReVelle, C., Bergman, L., 1971. The location of emergency service facilities. Oper. Res. 19 (6), 1363–1373.

Vance, P.H., 1998. Branch-and-price algorithms for the one-dimensional cutting stock problem. Comput. Optim. Appl. 9 (3), 211–228. doi:10.1023/A:1018346107246.

Vance, P.H., Barnhart, C., Johnson, E.L., Nemhauser, G.L., 1994. Solving binary cutting stock problems by column generation and branch-and-bound. Comput. Optim. Appl. 3 (2), 111–130. doi:10.1007/BF01300970.

Vanderbeck, F., Wolsey, L.A., 2010. Reformulation and decomposition of integer programs. In: Jünger, M., Liebling, T.M., Naddef, D., Nemhauser, G.L., Pulleyblank, W.R., Reinelt, G., Rinaldi, G., Wolsey, L.A. (Eds.), 50 Years of Integer Programming 1958–2008. Springer, pp. 431–502. doi:10.1007/978-3-540-68279-0_13.

Wallop, H., 2011. Japan earthquake: how Twitter and Facebook helped, The Telegraph. (accessed 9 September 2019). https://www.telegraph.co.uk/technology/twitter/8379101/Japan-earthquake-how-Twitter-and-Facebook-helped.html.

Welzl, E., 1991. Smallest enclosing disks (balls and ellipsoids). In: Maurer, H. (Ed.), New Results and New Trends in Computer Science. Springer, pp. 359–370.

Wu, Q., Zeng, Y., Zhang, R., 2018. Joint trajectory and communication design for multi-UAV enabled wireless networks. IEEE Trans. Wirel. Commun. 17 (3), 2109–2121.

Zeng, Y., Xu, X., Zhang, R., 2018. Trajectory design for completion time minimization in UAV-enabled multicasting. IEEE Trans. Wirel. Commun. 17 (4), 2233–2246.

Zeng, Y., Zhang, R., Lim, T.J., 2016. Throughput maximization for UAV-enabled mobile relaying systems. IEEE Trans. Commun. 64 (12), 4983–4996.

Zhan, C., Zeng, Y., Zhang, R., 2018. Energy-efficient data collection in UAV enabled wireless sensor network. IEEE Wirel. Commun. Lett. 7 (3), 328–331.

Zorbas, D., Di Puglia Pugliese, L., Razafindralambo, T., Guerriero, F., 2016. Optimal drone placement and cost-efficient target coverage. J. Netw. Comput. Appl. 75, 16–31. doi:10.1016/J.JNCA.2016.08.009.