공학박사학위논문

Large-scale Optimization Techniques for Unmanned Aerial Vehicle Operation Considering Covering Models

무인항공기 운영을 위한 덮개 모델 기반의 대규모 최적화 기법

2021 년 2 월

서울대학교 대학원

산업공학과

박 영 수

Large-scale Optimization Techniques for Unmanned Aerial Vehicle Operation Considering Covering Models

무인항공기 운영을 위한 덮개 모델 기반의 대규모 최적화 기법

지도교수 문 일 경

이 논문을 공학박사 학위논문으로 제출함

2020 년 12 월

서울대학교 대학원

산업공학과

박 영 수

박영수의 공학박사 학위논문을 인준함

2021 년 1 월

위 원 장	이 덕 주	(인)
부위원장	문 일 경	(인)
위 원	박 우 진	(인)
위 원	최 인 찬	(인)
위 원	강 준 규	(인)

Abstract

Large-scale Optimization Techniques for Unmanned Aerial Vehicle Operation Considering Covering Models

Youngsoo Park
Department of Industrial Engineering
The Graduate School
Seoul National University

There is increasing interest in the unmanned aerial vehicle (UAV) in various fields of the industry, starting from the surveillance to the logistics. After introducing the smart city, there are attempts to utilize UAVs in the public service sector by connecting individual components of the system with both information and physical goods. In this dissertation, the UAV operation problems in the public service sector is modeled in the set covering approach. There is a vast literature on the facility location and set covering problems. However, when operating UAVs in the system, the plan has to make the most of the flexibility of the UAV, but also has to consider its physical limitation. We noticed a gap between the related, existing approaches and the technologies required in the field. That is, the new characteristics of the UAV hinder the existing solution algorithms, or a brand-new approach is required.

In this dissertation, two operation problems to construct an emergency wireless network in a disaster situation by UAV and one location-allocation problem of the

UAV emergency medical service (EMS) facility are proposed. The reformulation to the extended formulation and the corresponding branch-and-price algorithm can overcome the limitations and improve the continuous or LP relaxation bounds, which are induced by the UAV operation.

A brief explanation of the UAV operation on public service, the related literature, and the brief explanation of the large-scale optimization techniques are introduced in Chapter 1, along with the research motivations and contributions, and the outline of the dissertations. In Chapter 2, the UAV set covering problem is defined. Because the UAV can be located without predefined candidate positions, more efficient operation becomes feasible, but the continuous relaxation bound of the standard formulation is weakened. The large-scale optimization techniques, including the Dantzig-Wolfe decomposition and the branch-and-price algorithm, could improve the continuous relaxation bound and reduce the symmetries of the branching tree and solve the realistic-scaled problems within practical computation time. To avoid numerical instability, two approximation models are proposed, and their approximation ratios are analyzed. In Chapter 3, UAV variable radius set covering problem is proposed with an extra decision on the coverage radius. While implementing the branch-and-price algorithm to the problem, a solvable equivalent formulation of the pricing subproblem is proposed. A heuristic based on the USCP is designed, and the proposed algorithm outperformed the benchmark genetic algorithm proposed in the literature. In Chapter 4, the facility location-allocation problem for UAV EMS is defined. The quadratic variable coverage constraint is reformulated to the linear equivalent formulation, and the nonlinear problem induced by the robust optimization approach is linearized. While implementing the large-scale optimization techniques, the struc-

ture of the subproblem is analyzed, and two solution approaches for the pricing subproblem are proposed, along with a heuristic.

The results of the research can be utilized when implementing in the real applications sharing the similar characteristics of UAVs, but also can be used in its abstract formulation.

Keywords: branch-and-price, column generation, emergency wireless network, emergency medical service, facility location problem, location-allocation problem, robust optimization, set covering problem, unmanned aerial vehicle

Student Number: 2014-21815

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Unmanned aerial vehicle operation on public services

Various fields of industry are showing an increasing interest in unmanned aerial vehicles (UAV). The main advantage of UAVs is their autonomous swarm operation capability [67, 68], 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.

The introduction of the UAV started from the military and surveillance area, but now is in the most active spotlight in the logistics applications. Along with the various unmanned systems, the UAV is becoming an essential component of the last-mile logistics of the next generation.

At the same time, there are attempts to utilize the UAV in the public service sector. UAVs can be operated as multipurpose resources providing rapid and flexible responses. Specifically, after introducing the smart city, the inspection and the maintenance of the infrastructure, monitor of the traffic condition, monitor of the vulnerable security points, and the disaster identification have been investigating the possibilities of the UAVs. UAVs can become a link connecting the physically or

socially remote area to the central authorities.

In the case of unexpected situations, the advantages of the UAVs become more prominent. The UAV can access the remote area without interruption to the road situation and react to the rapidly changing environment. Especially for disaster management, there are many projects to utilize UAV for the routing-related tasks and the covering-related tasks.

Meanwhile, attempts are trying to implement UAVs to search for and identify patients or transport emergency medicine, blood, and automated external defibrillator (AED). It is challenging to maintain a responsive EMS system in remote areas, such as hamlets and isolated dwellings, and in city centers with heavy traffic [29]. However, if UAVs can augment the EMS system, they can be used to access demand points through the air, thereby avoiding traffics.

In this dissertation, we are modeling the public service sector's UAV operation problems based on the set covering approach. Two types of problems are considered in the research. The first problem is to use UAVs to construct an emergency wireless network in a disaster situation. UAVs can cover certain areas and communicate with the sensors and the survivors, which can be modeled as a set covering problem with a coverage distance. The second problem is the facility location of the emergency medical service (EMS) operated by the AED-mounted UAVs. The problem locates the EMS facility along with the number of UAVs assigned to it. Based on the number of UAVs, the facility's capability is decided, which is related to its capacity and the coverage distance.

When planning the UAV operation system, the advantages and the limitations of the UAV have to be considered. Furthermore, if those characteristics hinder the

direct usage of the existing literature, a new optimization model has to be designed.

1.2 Facility location problems

There are three approaches in the facility location problems related to this research. The first is the continuous location approach. The continuous location problem, including the minimum covering circle problem, aims to decide the position of facilities in the xy-plane with various types of distance (e.g., Euclidean, rectangular, and p-norms). While deciding the position of given number of facilities, 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. [40] and Plastria [85] provide detailed information about the continuous location problem approach.

The second is the set covering-based approach. 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 binary feasibility of each facility and demand point pair. In the covering-based approach, the actual distance is binarized, and the demand point is considered to be "covered" if it is in the critical coverage distance from a facility.

The third approach is the median-based approach. In the median-based approach, the distance between the facility and the demand point is considered as a cost in the objective function instead of a constraint. In other words, usually there is no limitation on the coverage distance of a facility. Any demand point can be assigned to a facility as a decision, and the distance-weighted cost is accumulated to the objective function in the company with the facility's opening cost. Despite of

the pioneering literature considering maximum distance constraints in median-based approach as Toregas et al. [101] and Choi and Chaudhry [28], the objective of the median-based approach usually focuses on the total operation cost related to the summation of the distances between every demand point and the allocated facility.

The main difference between the last two approaches is based on the way the distance between a facility and a demand point is considered. In the covering-based approach, the distance is binarized with the coverage distance limitation of a facility. Conversely, in the median-based approach, the demand point is allocated without any restriction, even though the cost of the allocation can be high-valued.

1.3 Large-scale optimization techniques

Mixed-integer linear optimization heavily depends on the linear programming (LP) based branch-and-bound and the cutting plane method. The LP-based branch-and-bound is based on the *divide-and-conquer*, which provides a dual bound and the global best feasible (integer) solution by solving the LP relaxation of the original problem. On the other hand, the branch-and-cut algorithm generates the *valid inequality* from the fractional solution of the LP relaxation to improve the LP bound. The commercial optimization solver utilizes both approaches, where the root node LP is strengthened by adding the valid inequalities and then solved by the branch-and-bound.

However, when solving large-scale problems, there is a decomposition-based approach that can be helpful for reducing the variables by handling them implicitly, which can improve the LP bound of the problem. Minkowski's theorem proves that a polyhedron can be represented by its extreme points and extreme rays instead of the

original variables. That is, a vector in a polyhedron can be represented as a summation of a convex combination of the extreme points and a conic combination of the extreme rays of the polyhedron. Dantzig-Wolfe decomposition reformulates the original problem by decomposing the block-diagonal structure of the constraint into the smaller subproblems and the extended formulation [31]. Each subproblem consists of the structural constraints, and the resulting variables in the extended formulation implicitly satisfies those structural constraints. Minkowski's theorem enables Dantzig-Wolfe decomposition. By definition, even when the extended formulation is linear programming relaxed, the solution satisfies the structural constraints so can provide tighter bounds than the LP bound of the original formulation. Exactly speaking, the LP relaxation of the extended formulation is the dual of the Lagrangian subproblem. Thus, the LP bound of the extended formulation has the same value of the Lagrangian dual bound.

Instead of enumerating the exponential number of possible variables of the extended formulation, the column generation helps to manage the variable efficiently. By finding a new variable that improves the restricted master linear problem with iterations, only the required variables to represent the feasible set is generated and managed.

Another advantage of the branch-and-price algorithm is that if the decomposed subproblem have the structures of the well-studied problems, the techniques for those known problems can be utilized to solve the problem. Based on these advantages, the branch-and-price algorithm has been successfully implemented to handle cutting stock problem [104, 103], bin packing problem [90], vertex coloring problem [77], clique partition [78, 62], vehicle routing problem [36], crew scheduling problem [37],

and generalized assignment problem [93]. In this problem, the column generation and the corresponding branch-and-price approach are used to solve the proposed problems.

1.4 Research motivations and contributions

The planning model of the UAV operation system has to consider the characteristics of the UAV, which provides an additional degree of freedom and additional constraints, also. The first and the second problems in this dissertation locate individual UAVs to provide emergency wireless networks to the disaster scene, based on the set covering approach. Unlike the traditional discrete facility location problems, the UAVs can locate without a candidate position that limits the global optimal operation. However, because of the payload limitation, only a bounded area can be covered by the UAV-enabled emergency wireless network.

The third problem in this dissertation locates the EMS facility operating UAV EMS. The flight distance limitation of a UAV provides a hard constraint on the coverage distance around a facility. The problem decides both facility location and the assignment of the UAV to each facility, which decides the capacity and the coverage distance of the facility.

We denote the first problem as UAV set covering problem (USCP). The USCP has two characteristics, which are distinct from the existing literature. The objective of the USCP is to cover every demand point with a minimum number of UAVs. Thus, the problems with a given number of resources, the continuous location problem, and the clustering-based approach are not suited for the USCP. On the other hand, because the UAV can fly at any point, the discrete facility location problems,

including the nominal set covering and location-allocation problems, can only provide local optimal solutions. Because of the characteristics of the problem, a new modeling approach is required. Although the USCP has a fairly straightforward definition, the problem requires a substantial computational burden, which originates from an exponential number of subsets to be checked (see Chapter 2).

The second problem, UAV variable radius set covering problem (UVCP) considered the extra decision on the coverage radius of each UAV. The USCP assumes the homogeneous radius of the UAV network. In real applications, the radius of the network can be changed when the UAV flights at different altitudes. The allowance of the extra decision results in the nonconvex feasible set of the continuous relaxation, which makes the problem impossible to be solved by an optimization solver. Thus, Berman et al. [14] claimed that the UVCP requires the brute force search to find the optimal solution. However, the problem can be modeled as a solvable mathematical model, using the knowledge of the minimum covering circle (see Chapter 3).

The third problem is denoted as facility location-allocation problem for UAV EMS (ULAP). In the ULAP, the limitation of the flight distance requires a hard constraint of the coverage distance. In the literature considering the variable but bounded coverage distance, the resource capacity and the availability are not considered. On the other hand, in the nominal location-allocation problems, the resource availability is modeled as a cost parameter rather than a constraint. When considering the bounded coverage distance in the existing literature, the problem is modeled with a covering-based approach. In the ULAP, the coverage distance of a UAV EMS facility is modeled as a function of the number of the UAV. Because the ULAP includes a strategic decision of the long-term operation, the demand uncertainty is

considered, which also complicated the problem (see Chapter 4).

Figure 1.1 illustrates the overview of the research problems investigated in this dissertation. The dissertation considers two problem environments of disaster and ordinary situations. The USCP and the UVCP in Chapters 2 and 3 considers the planning problem of the UAV-based emergency wireless network in a disaster environment. As shown in the figure, while locating the UAVs in the disaster area, the USCP assumes the homogeneous UAVs, which provides the fixed-radius network coverage area, and the UVCP decides the size of the coverage area simultaneously. The ULAP in Chapter 4 decides the location and the capability of the UAV-based emergency medical service facility in an ordinary situation. Based on the number of the UAVs assigned to each facility, the coverage area and the capacity of the facility is decided.

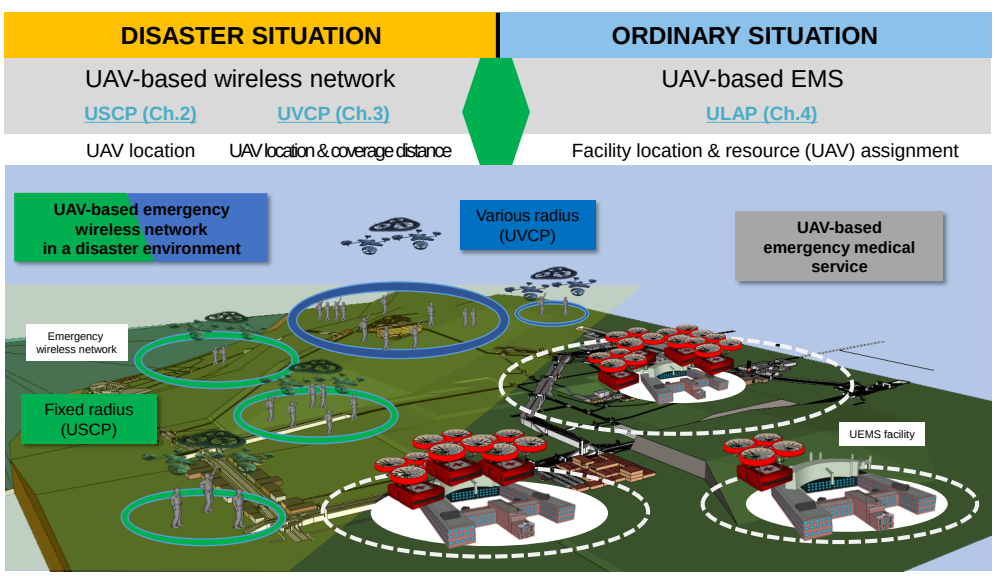


Figure 1.1: Overview of the dissertation

The comparison of the existing modeling approaches and the researches in this dissertation is summarized in Table 1.1.

Table 1.1: Comparison of this dissertation and existing approaches

Chapter	**Research problem**	**Candidate position**	**Number of resources**	**Coverage distance constraint**
2 & 3	Continuous facility location	without	fixed	fixed
2 & 3	Covering-based approach	given	not fixed	fixed
2 & 3	Median-based approach	given	not fixed	not fixed
2	This dissertation (USCP)	without	not fixed	fixed
3	This dissertation (UVCP)	without	not fixed	not fixed
Chapter	**Research problem**	**Candidate position**	**Demand**	**Coverage distance constraint**
4	Covering-based approach, Median-based approach	given	deterministic, partial, uncertain	double, backup, gradual, variable
4	This dissertation (ULAP)	given	uncertain	variable

The principal contributions of the dissertation are summarized as follows:

1. For the UAV set covering problem,

    - The problem can be modeled as a set covering problem with an extra decision of the position of the UAV, which is a mixed-integer quadratically-constrained programming model.
    - The extra decision on the position neutralizes the continuous relaxation bound of the standard formulation to 1.
    - For Dantzig-Wolfe decomposition, the generating set of the USCP is designed for the quadratic coverage constraint and the continuous decision variable.

- By implementing Dantzig-Wolfe reformulation, the continuous relaxation bound is improved and could solve the realistic-scale problems within the practical computation time.
- For a numerical stability, the approximation of the problem, which is mixed-integer linear programming model, is proposed based on Jung's theorem [65].
- The approximation ratio of the approximation model is provided, along with an approximation model based on the discretization of every lattice point over the plane.
- The algorithmic performances of the proposed algorithms are analyzed.

2. For the UAV variable radius set covering problem,

   - The nonconvex continuous relaxation of the problem is shown.
   - The problem is reformulated to the extended formulation and solved by the branch-and-price algorithm.
   - By using the extended formulation, the concept of the minimum covering circle can be used, and the bound for the number of the columns is provided.
   - Based on the minimum covering circle, the equivalent subproblem, which is a mixed-integer linear programming model, is proposed. It allowed the UVCP to be reformulated by Dantzig-Wolfe decomposition and the pricing subproblem to be solved by the optimization solver.
   - Radius-fixing-based heuristic is designed and implemented to the hybrid approach of the heuristic and the branch-and-price algorithm.

- The proposed algorithms outperformed the genetic algorithm proposed in Berman et al. [14].

3. For the facility location-allocation problem for UAV EMS,

   - A cost-minimization problem is developed based on the uncertain demand with a robust optimization approach with a cardinality-constrained uncertainty set.
   - The number of UAVs assigned to a UAV EMS facility is modeled to be related to both capacity and the coverage distance constraint, which is introduced for the first time in this research.
   - To apply the strict limitation of the flight distance constraint, the variable coverage distance constraint is modeled. Based on the proximity of the resource availability and the size of the covered area, the variable coverage distance constraint is modeled as a quadratic constraint and linearized.
   - The integer and continuous decision variables of the model have highly fractional LP solutions and weak LP bound. A branch-and-price algorithm is introduced for the better LP bound, and the appropriate branching strategy is designed.
   - The subproblem is analyzed to be the robust disjunctively constrained integer knapsack problem. Two approaches of mixed-integer linear programming (MILP) reformulation and decomposed dynamic programming approaches are proposed to solve the subproblem, and the advantages are analyzed.

- A restricted master heuristic based on the B&P algorithm is proposed to provide a time-efficient feasible solution to large-sized problems.

The USCP and the UVCP in Chapters 2 and 3 allow the continuous decision of the location and therefore improves the existing set covering approach of the facility location problem. The same approach can be implemented in various applications considering the clusters such as data classification, bandwidth packing, location-routing, and generalized traveling salesman problems. The ULAP in Chapter 4 relates the variable coverage distance to the resource availability, which can be utilized when the variable coverage distance and the capacity of the facility is decided based on the resource assigned. The solution algorithms proposed in the research can help solving various types of facility location-allocation problems.

1.5 Outline of the dissertation

In this dissertation, we consider three problems related to the UAV operation with covering models. In Chapters 2 and 3, the location problems of UAVs to provide emergency network to the disaster scene are introduced. In Chapter 4, the facility location-allocation problem based on the UAV assignment is introduced. In Chapter 2, the UAV set covering problem is defined, and the weak continuous relaxation bound of the standard formulation is shown. The extended formulation and the corresponding branch-and-price algorithm is proposed, which could solve the realistic-scaled problems within reasonable computation time. To avoid the numerical instability which is caused by the quadratic constraint, a linear approximation model is designed based on the Jung's theorem. In Chapter 3, UAV variable radius set covering problem is proposed. An explicit mathematical model is proposed

with the analysis of the feasible region of the problem. While implementing the branch-and-price algorithm to the problem, a solvable equivalent mixed-integer linear formulation of the pricing subproblem is proposed. A heuristic based on the USCP is designed with a hybrid, exact algorithm for the fast computation speed. In Chapter 4, we define the facility location-allocation problem for UAV EMS. The quadratic variable coverage constraint is reformulated to the linear equivalent formulation, and the nonlinear problem induced by the robust optimization approach is linearized. The extended formulation and the corresponding branch-and-price algorithm is designed, and the branching strategies are compared. The structure of the subproblem is analyzed, and two solution approaches for the pricing subproblem are proposed. In addition, a heuristic based on the extended formulation is proposed for the time-efficient solution of the large-sized problems, and the performances of the algorithms are compared. Finally, Chapter 5 concludes the dissertation.

Chapter 2

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

2.1 Introduction

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 [108]. Besides, some survivors used the wireless network to inform authorities of the current immediate situation and to request rescue [2]. Similarly, survivors of the 2010 Haiti earthquake [52] and Hurricane Harvey in southeast Texas [56] 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, such as a visual surveillance and monitoring system based on camera-mounted UAVs that need to operate near the actual site. Ho et al. [53] used particle swarm optimization as an approximation algorithm to optimize the UAV trajectory along waypoint candidates. Zhan et al. [116] synchronized the wake-up schedule of sensors and the UAV trajectory. Zeng et al. [115] and Wu et al. [112] maximized the throughput controlling trajectory and speed of the UAV. Zeng et al. [114] 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 chapter. In this case, the problem can be modeled based on the facility location problem, which is introduced in Section 1.2. In the set covering-based approach, Chandrashekar et al. [27] modeled a two-layer network that consists of mobile ad hoc networks (MANET) and a

covering UAV network. Zorbas et al. [122] proposed a minimum-cost drone location problem that takes into consideration the network coverage changing over flight altitudes. In the continuous facility location problem, some of the considerable research has focused on the clustering problems such as K-means [83, 92], modified K-means [83], disk covering [80], and circle packing in a circle [79]. In the median-based approach of the UAV location, there were p-center [34] and p-cover [23] problems, which also considered the given number of the clusters p as the primary constraint instead of the physical limitation of the coverage radius.

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 facility location problem and median-based approach 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. [19] 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 continuous relaxation bound for the USCP does not provide any information, as explained in Section 2.3.1. Note that the solution algorithm of p-center problem has a complexity of $O(|N|^p)$ [24]. 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- and median- based approaches decides among given

Table 2.1: Comparison of this research and existing literature

Author (year)	**Type**	**Given candidate position**	**Fixed number of resources**
Comley (1995) [30]	Continuous	No	Yes
Drezner et al. (2001) [40]	Continuous	No	Yes
Plastria (2001) [85]	Continuous	No	Yes
Capoyleas et al. (1991) [24]	Continuous	No	Yes
Sasikumar and Khara (2012) [92]	Continuous	No	Yes
Periyasamy et al. (2016) [83]	Continuous	No	Yes
Daskin (1983) [32]	Set cover	Yes	No
Gendreau et al. (1997) [47]	Set cover	Yes	No
Zorbas et al. (2016) [122]	Set cover	Yes	No
ReVelle and Hogan (1989) [88]	Set cover	Yes	Yes
Choi and Chaudhry (1993) [28]	Median	Yes	Yes
Calik et al. (2015) [23]	Median	No	Yes
Daskin and Maass (2015) [34]	Median	No	Yes
Wankmüller et al. (2020) [109]	Median	Yes	No
This dissertation (Chapters 2 & 3)	Set cover	No	No

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 [101], the 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 [21, 5, 1, 10]. 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. Table 2.1 compares this research to the existing literature.

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 chapter 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 and allowed to be not connected each other 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; [104, 103, 63, 62]). The reformulation associated with the B&P algorithm strengthens the continuous relaxation 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 USCP has a basic structure of the set covering problem, in which the physical location and distribution of the demand points over the plane plays a significant role in the solution. Because the location of the UAV is not limited to the given candidate locations in the USCP, more efficient solutions can be provided. The approach is not confined to the UAV-related problems but also can be used in various operation problems. For the problems that the physical locations and the distance constraints are important, and continuous decision variables over the space are incorporated share the similar structures of the USCP. In the problems of the data clustering, if the diameter of each cluster is important than limiting the fixed number of the cluster, the approach of the USCP would be more suited than the support vector machine or the K-means. If the information of the multidimensional data is given and the distance between the data points is defined, the USCP can be used to cluster the data into the minimum number of subsets, where each subset is bounded within a given diameter. In this approach, the outliers would be identified more intuitively. Furthermore, other problems of bandwidth packing, location-routing, and generalized traveling salesperson problems can get advantages from the USCP, too.

For the technical implication of the research, the coverage distance constraint is modeled as a quadratic equation, which makes the problem to be MIQCP. Despite of the continuous decision variable and the quadratic constraint, the USCP can be decomposed by the generic framework of Dantzig-Wolfe reformulation, and the gen-

erating set can be defined based on the projection to the discrete decision variable. It implies that similar problems can be solved by Dantzig-Wolfe decomposition and the corresponding B&P algorithm. Apart from the USCP, the problems that decide the location freely on the xy-plane as the continuous decision can be approached by this solution procedure.

The rest of this chapter is structured as follows: Section 2.2 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 2.3 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 2.4 presents the computational experiments conducted, including the algorithmic performances of four proposed models. In this section, we analyze the managerial insights for practical applications in a disaster environment. Section 2.5 introduces the structure of the solution and the application of the USCP. Finally, Section 2.6 concludes the research.

2.2 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.

In this research, the set covering problem does not consider the network coverage distance from the ground access point to the UAV or between the UAVs. The wireless network that is provided by the UAV to cover the area (UAV-to-ground wireless network) is omnidirectional and has a limited transmission capacity and the limited coverage distance. However, as considered in Khan et al. [66], UAV-to-UAV and UAV-to-gateway networks are more likely directional transmissions and therefore have a longer transmission distance. Based on these characteristics, in the USCP, the UAVs are assumed to be connected to the authority, regardless of their location.

On the other hand, even in the problem environment in which the distance between the UAVs or the UAV-to-gateway transmission distance provides constraints, the USCP model and the solution can give a critical input. In the cases that utilize the multi-tier network architecture [95, 69] or the connected set covering approach that limits the communication distance between the UAVs, the solution of the USCP can be used. As explained, this research investigates the simplest model of the UAV

set covering problem to identify the topographical characteristic of the problem. The ways to utilize the USCP in more practical problems are presented in Section 2.5.

2.2.1 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.

Figure 2.1 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 chapter 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.

2.2.2 Mathematical formulation

Based on the problem defined in Section 2.2.1, a mathematical model is developed. The followings are the notations used in the standard mathematical formulation for the USCP:

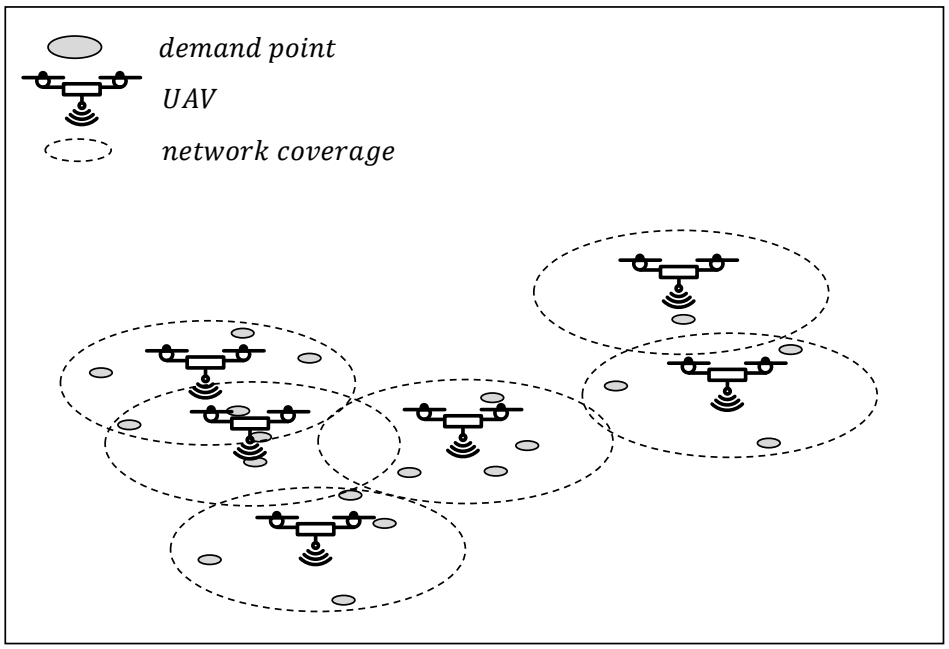


Figure 2.1: Overview of the USCP

Set

N	set of demand points.

Parameters

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

Decision variables

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

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

$$c_j^{\hat{x}} \in \mathbb{R}, \quad \text{position of UAV } j \text{ on x-coordinate.} \qquad \forall j \in \{1, ..., |N|\}$$

$$c_j^{\hat{y}} \in \mathbb{R}, \quad \text{position of UAV } j \text{ on y-coordinate.} \qquad \forall j \in \{1, ..., |N|\}$$

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, which has a cardinality of $|N|$. The following is the standard mathematical formulation of the USCP. For distinction, the formulation will be renamed as Euclidean standard (ES) formulation.

$$\min \quad \sum_{j=1}^{|N|} y_j \tag{2.1}$$

$$\text{s.t.} \quad x_{ij} \leq y_j, \qquad \forall i \in N, \forall j \in \{1, ..., |N|\} \tag{2.2}$$

$$\sum_{j=1}^{|N|} x_{ij} \geq 1, \qquad \forall i \in N \tag{2.3}$$

$$(a_i^{\hat{x}} - c_j^{\hat{x}})^2 + (a_i^{\hat{y}} - c_j^{\hat{y}})^2 \leq R^2 + \tilde{M}(1 - x_{ij}), \quad \forall i \in N, \forall j \in \{1, ..., |N|\} \tag{2.4}$$

$$x_{ij} \in \mathbb{B}, \qquad \forall i \in N, \forall j \in \{1, ..., |N|\} \tag{2.5}$$

$$y_j \in \mathbb{B}, \qquad \forall j \in \{1, ..., |N|\} \tag{2.6}$$

$$c_j^{\hat{x}}, c_j^{\hat{y}} \in \mathbb{R} \qquad \forall j \in \{1, ..., |N|\} \tag{2.7}$$

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 (2.2) is a linking constraint between a demand point and a UAV; the deployment of a UAV precedes the assignment of a demand point. Constraint (2.3) is a demand assignment constraint; every demand is required to be fulfilled by at least one UAV. Constraint (2.4) 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, $(a_i^{\hat{x}}, a_i^{\hat{y}})$ and the position of UAV j, $(c_j^{\hat{x}}, c_j^{\hat{y}})$ is less than the coverage radius R. Constraints (2.5), (2.6), and (2.7) define the dimension of the decision variables. The quadratic shape of Constraint (2.4) originates in Comley [30]. 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.

2.2.3 Discrete approximation model

Constraint (2.4) is quadratic because of the continuous decision variable for the position of the UAV, $c_j^{\hat{x}}$ and $c_j^{\hat{y}}$. 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

$b_j^{\hat{x}}$ flight position of UAV j on x-coordinate. $\forall j \in M$

$b_j^{\hat{y}}$ 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

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

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

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 $\sqrt{(a_i^{\hat{x}} - b_j^{\hat{x}})^2 + (a_i^{\hat{y}} - b_j^{\hat{y}})^2} \leq R$ and 0 otherwise. Except for the domain of the binary decision variable y_j and the discretization of the continuous decision variables $c_j^{\hat{x}}$ and $c_j^{\hat{y}}$, the mathematical formulation of the DA model is almost identical to that of the ES model:

$$\min \quad \sum_{j \in M} y_j \tag{2.8}$$

$$\text{s.t.} \quad x_{ij} \leq y_j, \qquad \forall i \in N, \forall j \in M \tag{2.9}$$

$$\sum_{j \in M} \alpha_{ij} x_{ij} \geq 1, \qquad \forall i \in N \tag{2.10}$$

$$x_{ij} \in \mathbb{B}, \qquad \forall i \in N, \forall j \in M \tag{2.11}$$

$$y_j \in \mathbb{B}, \qquad \forall j \in M \tag{2.12}$$

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 2.3.4 and 2.4.

2.3 Branch-and-price approach for the USCP

Section 2.3.1 presents an extended formulation for the B&P algorithm. Section 2.3.2 introduces a detailed branching strategy related to the B&P algorithm on the USCP. Section 2.3.3 examines Jung's theorem and the approximation model with pairwise-conflict constraints based on Jung's theorem. Section 2.3.4 compares two approximation models based on the approximation ratio. Section 2.3.5 presents the overall algorithmic framework to use pairwise-conflict constraint approximation model in a disaster situation.

2.3.1 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 [100]. However, the continuous relaxation of the USCP neutralizes the coverage constraint. Unlike in the nominal set covering problem, the coverage constraint is considered jointly by Constraints (2.3) and (2.4) in the ES formulation. The continuous relaxation separates the relation among x_{ij}, $c_j^{\hat{x}}$, and $c_j^{\hat{y}}$. Thus, the solution of the continuous 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 bound as 1.

We propose a Dantzig-Wolfe decomposition based on each UAV. Because the USCP includes the continuous decision variables, $c_j^{\hat{x}}$ and $c_j^{\hat{y}}$, the concept of generating sets proposed in Vanderbeck and Savelsbergh [106] is used. While discretizing the binary variables, x_{ij} and y_j, generating sets for the subsystem of a UAV can be defined and reformulate the USCP. Let X^B be a subsystem for a UAV:

$$\begin{aligned} X^B = \big\{ (\boldsymbol{c}, \boldsymbol{x}, \boldsymbol{y}) : (a_i^{\hat{x}} - c_j^{\hat{x}})^2 + (a_i^{\hat{y}} - c_j^{\hat{y}})^2 \leq R^2 + \tilde{M}(1 - x_{ij}), x_{ij} \leq y_j, \\ \forall i \in N, \forall j \in \{1, ..., |N|\}; \boldsymbol{c} \in \mathbb{R}_+^{|N|\times|N|}; \boldsymbol{x} \in \mathbb{B}^{|N|\times|N|}; \boldsymbol{y} \in \mathbb{B}^{|N|} \big\}, \end{aligned}$$

where $\boldsymbol{c} = \big(\boldsymbol{c_1}, ..., \boldsymbol{c}_{|\boldsymbol{N}|}\big)$ and $\boldsymbol{c_j} = (c_j^{\hat{x}}, c_j^{\hat{y}})$. A set of the feasible flight positions of a

UAV, $S^B(\boldsymbol{x}, \boldsymbol{y})$, can be defined as a union of the circles:

$$S^B(\boldsymbol{x}, \boldsymbol{y}) = \big\{\boldsymbol{c} \in R^{|N|\times|N|} : (a_i^{\hat{x}} - c_j^{\hat{x}})^2 + (a_i^{\hat{y}} - c_j^{\hat{y}})^2 \leq R^2 + \tilde{M}(1 - x_{ij}), x_{ij} \leq y_j,$$
$$\forall i \in N, \forall j \in \{1, ..., |N|\}; [x_{ij}] = \boldsymbol{x}, [y_j] = \boldsymbol{y}; \boldsymbol{c} \in \mathbb{R}_+^{|N|\times|N|}\big\}.$$

The *generating set* G^B and its projection G_p^B can be defined as:

$$G_p^B = proj_{(x,y)} X^B = \big\{\boldsymbol{x} \in \mathbb{B}^{|N|\times|N|}; \boldsymbol{y} \in \mathbb{B}^{|N|} : (a_i^{\hat{x}} - c_j^{\hat{x}})^2 + (a_i^{\hat{y}} - c_j^{\hat{y}})^2 \leq R^2$$
$$+\tilde{M}(1 - x_{ij}), x_{ij} \leq y_j, \forall i \in N, \forall j \in \{1, ..., |N|\}; \boldsymbol{c} \geq 0\big\},$$
$$G^B = \big\{(\boldsymbol{c}, \boldsymbol{x}, \boldsymbol{y}) \in \mathbb{R}_+^{|N|\times|N|} \times \mathbb{B}^{|N|\times|N|} \times \mathbb{B}^{|N|} : (\boldsymbol{x}, \boldsymbol{y}) \in G_p^B; \boldsymbol{c} \in S^B(\boldsymbol{x}, \boldsymbol{y})\big\}.$$

The reformulation based on the proposed generating set G^B can be defined as:

$$\begin{aligned}
\min \quad & \sum_{g \in G^B} \sum_{j=1}^{|N|} y_j^g \lambda_g & \\
\text{s.t.} \quad & \sum_{g \in G^B} \sum_{j=1}^{|N|} x_{ij}^g \lambda_g \geq 1 & \forall i \in N \\
& \lambda_g \in \mathbb{B}, & \forall g \in G^B
\end{aligned}$$

where λ_g denotes the weight of the generator g and $(\boldsymbol{c^g}, \boldsymbol{x^g}, \boldsymbol{y^g})$ denotes the solution $(\boldsymbol{c}, \boldsymbol{x}, \boldsymbol{y})$ defined by generator g. However, one can use a finite generating set G_p^B for the reformulation. The reformulation cannot fully describe the feasible set but the projection of the original problem. However, the objective function can be described by the variables x_{ij} and y_j, thus the decomposition can be executed based on the projection over $(\boldsymbol{x}, \boldsymbol{y})$. We first define the generator $k \in G_p^B$. Furthermore, there is no predefined information related to the UAV, so that one can replace the index of

the UAV j with the generator k. Thus, the generator k can be defined solely based on the variable $\boldsymbol{x}$. The term "column" will be used instead of the generator, and the "set of columns" will be used instead of the generating set for the convenience.

In the extended formulation, the fixed-radius coverage constraint with Euclidean distance (2.4) is considered implicitly in the decision variable; therefore, the solutions of the continuous relaxation satisfy the coverage constraints, which obtains tighter continuous relaxation bounds than in the nominal set covering problem. Another advantage of the decomposition is the elimination of symmetry among solutions, which obstructed the search on the B&B algorithm [107]. 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 $k_1 = \{1, 3\}$, $k_2 = \{2, 3\}$, and $k_3 = \{1, 2\}$ are considered, one can cover demand points $\{1, 2, 3\}$ by selecting columns k_1 and k_2, or k_2 and k_3. Let a generating set Ω be the set of every feasible column that implicitly represents a set of demand points covered by one UAV. For each assignment pattern $k \in \Omega$, the inclusion of each demand point i is defined as a binary parameter w_{ik}. A binary decision variable z_k is defined for each feasible column to denote the adoption. The extended formulation model of the USCP is represented in the following integer program:

$$\min \quad \sum_{k \in \Omega} z_k \tag{2.13}$$

$$\text{s.t.} \quad \sum_{k \in \Omega} w_{ik} z_k \geq 1 \qquad \forall i \in N \tag{2.14}$$

$$z_k \in \mathbb{B} \qquad \forall k \in \Omega \tag{2.15}$$

Objective function (2.13) minimizes the cardinality of UAVs operated to cover every demand point. Constraint (2.14) 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_k because the size of the set Ω is exponential on the number of demand points $|N|$. 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 solution of the root node CG does not satisfy the integrality. Let π_i be a dual price associated with constraint (2.14); additional columns for the restricted master problem can be generated by solving the following pricing problem:

Decision variables

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

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

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

$$\min \quad 1 - \sum_{i \in N} \pi_i x_i \tag{2.16}$$

$$\text{s.t.} \quad (a_i^{\hat{x}} - c^{\hat{x}})^2 + (a_i^{\hat{y}} - c^{\hat{y}})^2 \leq R^2 + \tilde{M}(1 - x_i), \qquad \forall i \in N \tag{2.17}$$

$$x_i \in \mathbb{B}, \qquad \forall i \in N \tag{2.18}$$

$$c^{\hat{x}}, c^{\hat{y}} \in \mathbb{R}, \tag{2.19}$$

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 k is set to be 1 for every UAV in Formulation (2.1). Thus, the pricing subproblem is identical for every UAV. The objective function (2.16) 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 (2.4) in the ES formulation, is considered in Constraint (2.17), 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.

2.3.2 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 [35]. The Ryan-Foster branching rule [89] is often used in the set partitioning problem [61]. 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_1} and x_{i_2} for subproblem, which represents the assignment of demand points i_1 and i_2 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 $v_{i_1 i_2}$ is calculated. For each pair of demand points, the value

of the fractional solution of a column $\tilde{z}_k$ is aggregated if both demand points are included:

$$v_{i_1 i_2} := \sum_{k \in \Omega, w_{i_1 k} = w_{i_2 k} = 1} \tilde{z}_k.$$

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.

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

The reformulation provides a better continuous relaxation bound and eliminates the symmetries in the branching tree. Furthermore, the advancement of the non-linear solver engine enables the solution algorithm to solve the pricing subproblem efficiently regardless of the quadratic constraints. In ES and extended formulations of the USCP, quadratic constraints (2.4) and (2.17) 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 $c^{\hat{x}}$ and $c^{\hat{y}}$ co-exist in Constraint (2.17). 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 $\tilde{M}$ 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 pre-solver 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 [44]. 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 [110] 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. In Chapter 3, the minimum covering circle-based formulation is defined.

Instead of the Welzl's algorithm, 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 [90], Gendreau et al. [48], and Manerba and Mansini [76] used conflict constraints to model predefined incompatibility between choices. Grötschel and Wakabayashi [49], Hoffman and Padberg [54], and Borndörfer and Weismantel [20] 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 [65] proposed an inequality between the diameter and the radius of the minimum enclosing ball of a set:

Theorem 2.1 (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*

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

that contains K.

In the case of the xy-plane ($n = 2$), according to Jung's theorem, a circle with $r \leq \frac{d(K)}{\sqrt{3}}$ 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 2.2. *Considering a compact set $K \subset \mathbb{R}^n$. For a given $R \in \mathbb{R}^1$, if $d(K) \leq \sqrt{3}R$, then there exists a closed ball with radius $r \leq R$.*

According to Lemma 2.2, 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 (2.4) and (2.17) 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 $d_{i_1 i_2}$ be the distance between demand points i_1 and i_2. Constraints (2.4) and (2.17) can be approximated into Constraints (2.20) and (2.21), respectively:

$$x_{i_1 j} + x_{i_2 j} - 2 + C_{i_1 i_2} \leq 0 \qquad \forall i_1, i_2 \in N, \forall j \in \{1, ..., |N|\} \tag{2.20}$$

$$x_{i_1} + x_{i_2} - 2 + C_{i_1 i_2} \leq 0 \qquad \forall i_1, i_2 \in N \tag{2.21}$$

where pairwise-conflict parameter $C_{i_1 i_2} := \dfrac{(d_{i_1 i_2})^2 - 3R^2}{(\max\limits_{i_1, i_2 \in N} [d_{i_1 i_2}])^2 - 3R^2}$. If $d_{i_1 i_2} \leq R\sqrt{3}$, Constraints (2.20) and (2.21) become redundant. Otherwise, $0 \leq C_{i_1 i_2} \leq 1$ and Constraints (2.20) and (2.21) 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). Both PCA and PCBP models are mixed-integer linear programming. Let $\boldsymbol{x}, \boldsymbol{y}$ be a feasible solution of a PCS model. According to Lemma 2.2, $\boldsymbol{x}$, $\boldsymbol{y}$ 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 (2.16) and Constraints (2.18) and (2.19). Constraint (2.21) replaces Constraint (2.17) to represent the approximated coverage constraint. In the CG subproblem, Constraint (2.21) 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 coexistence. 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., [49, 61, 62]) 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 [78], 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 chapter be-

cause 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 2.3.4 and 2.4.

2.3.4 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 2.3. *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 Figure 2.2.

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 $|\mathcal{F}^*| \geq |\mathcal{F}^{PCA}|/3$. $\square$

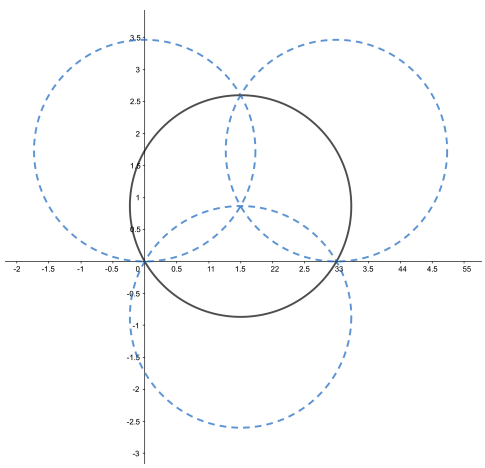


Figure 2.2: Worst-case of the PCA

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.

Theorem 2.4. *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 $3|\mathcal{F}^*| < |\mathcal{F}^{DA}_{G_d}|$ is suggested: Let $R = \sqrt{3}$ and $|N| = 4$ with $(a_i^{\hat{x}}, a_i^{\hat{y}}) = (\frac{17}{10}, \frac{\sqrt{11}}{5} - \frac{\sqrt{299}}{10}), (\frac{33}{10}, \frac{3\sqrt{11}}{10}), (\frac{17}{10}, \frac{\sqrt{11}}{5} + \frac{\sqrt{299}}{10})$, 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, $|\mathcal{F}^*| = 1$. In the DA model with $G_d = \sqrt{3}$, no lattice point that can cover more than one given demand point. Thus, $|\mathcal{F}^{DA}_R| > 3$ 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 2.4 compares performances of the DA and the PCA models with the exact model based on the computation speed and the objective value.

2.3.5 Framework of the solution algorithm for the PCBP model

Figure 2.3 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 (2.21) is calculated for each pair of demand points i_1 and i_2 based on the distance $d_{i_1 i_2}$ and $\tilde{M}$. 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 posi-

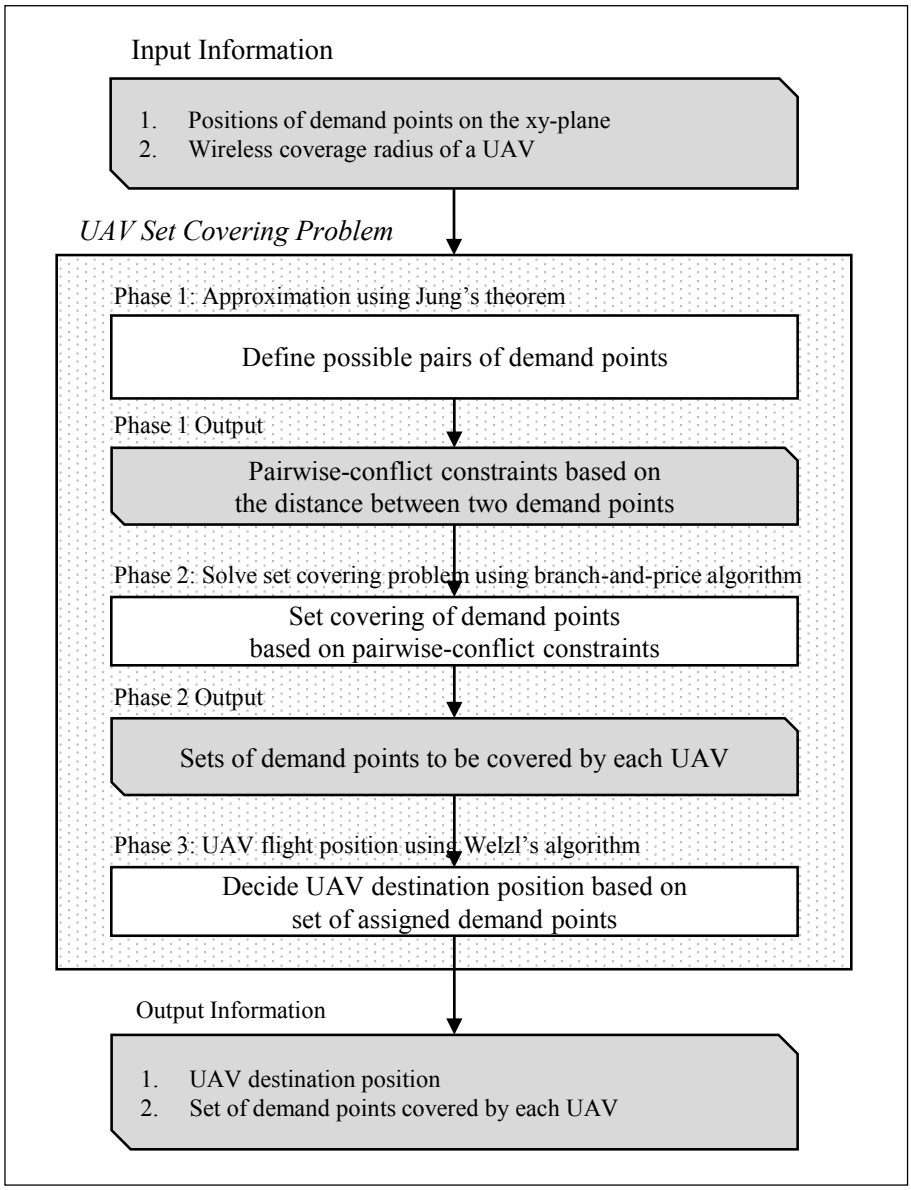


Figure 2.3: Framework of the solution algorithm for PCBP model

tion for each UAV. Therefore, an additional phase is required to decide the position 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. Figures 2.4a and 2.4b show solutions of the USCP, illustrated on the xy-plane.

2.4 Computational experiments

We conducted computational experiments to measure the performance of the proposed solution algorithms (Section 2.4.2). Section 2.4.1 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 3.60GHz and 24GB of RAM operated on a Windows 10 64-bit operating system.

2.4.1 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 capac-

itated p-median test problem in OR-Library [9], which was introduced by Osman and Christofides [81]. 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 $\times$ 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. [75]. 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 HKF dataset has been used in a variety of studies on disaster management [22, 64]. 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 meters to 1,000 meters [26, 50]. Thus, three radii—200, 1,000, and 2,000 meters—were tested for each instance. Figure 2.4b 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.

2.4.2 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 3,000 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. The DA can be seen as a benchmark algorithm in this research. The DA is the *fair* modeling considering the extra decision of the candidate location and depends solely on the performance of the optimization solver, which can help to assess the effectiveness of the other algorithms using the techniques tailored for the USCP. 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 2.2, 2.3, and 2.4 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.

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

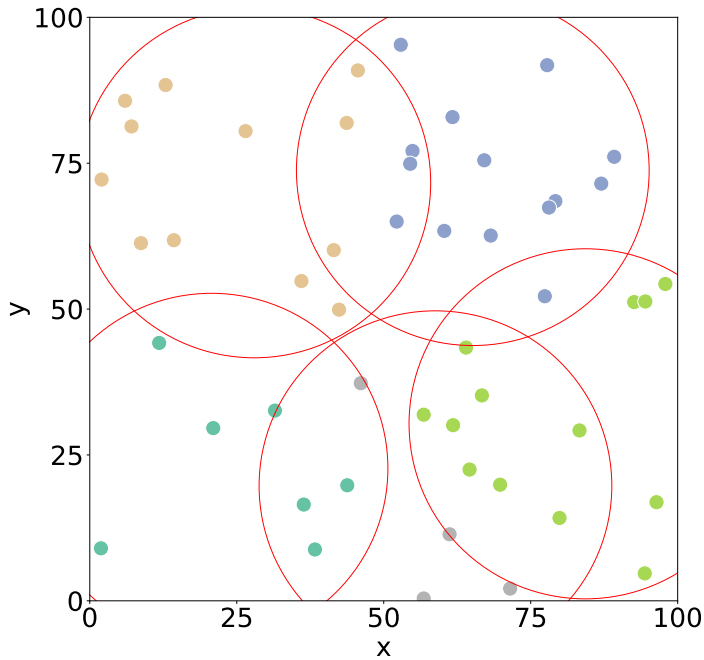
In the field of aerial operation, some circumstances demand sub-minute or sub-second time pressures. Table 2.2 shows that the EBP could solve almost every problem within 100 seconds, and the PCBP could solve every problem class except $|N| = 100$ and Rd =2,000, faster than the EBP. The computation speed 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, 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. Figure 2.4a 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

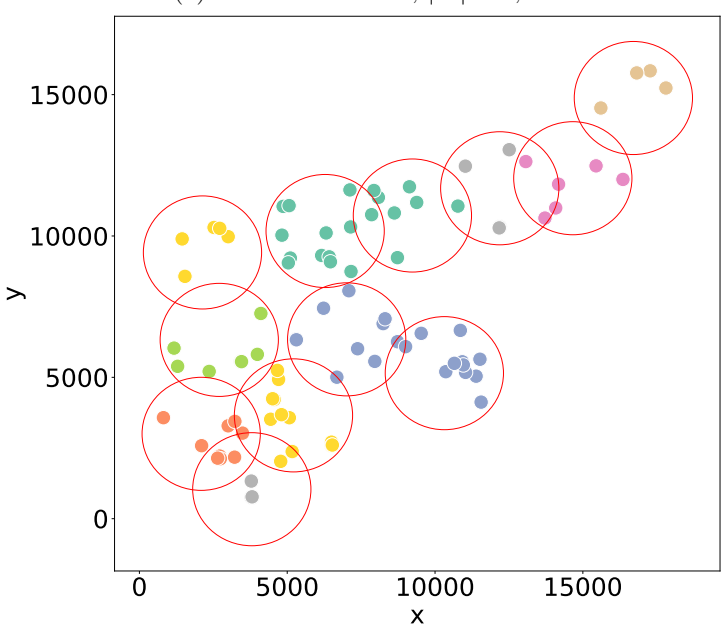
30. Thus, the solution does not change much as $|N|$ increases, which means that the concurrent optimizer of the commercial solver provides the solution fast.

Table 2.3 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 2.4 and 2.5 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 problems in which the DA showed shorter computation times and better 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.



(a) Artificial dataset, $|N|$=50, Rd=30



(b) Realistic-scale dataset, $|N|$=100, Rd=2000

Figure 2.4: Solution examples

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 2.3.4, the upper limit of grid size G_d is $\sqrt{2}Rd$. Table 2.4 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 2.5 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.

Table 2.2: 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	3000*	0.99	0.04	0.03	82.06	0.00	0.00	0.00
	20	0/10	10/10	10/10	10/10	3000*	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	3000*	10.39	393.99	2.02	96.00	0.00	0.77	0.00
	20	0/0	10/10	8/10	10/10	3000*	14.46	663.61	22.49	96.00	0.00	2.27	0.00
	30	0/8	10/10	10/10	10/10	3000*	8.35	3.15	70.02	68.05	0.00	0.00	0.00
50	200	0/0	10/10	10/10	10/10	3000*	3.10	0.79	0.18	95.80	0.00	0.00	0.00
	1000	0/0	10/10	10/10	10/10	3000*	6.69	110.15	1.08	95.60	0.00	0.00	0.00
	2000	0/0	10/10	10/10	10/10	3000*	7.07	95.14	7.38	95.75	0.00	0.00	0.00
100	200	0/0	10/10	10/10	10/10	3000*	32.02	360.90	3.72	98.98	0.00	0.00	0.00
	1000	0/0	10/10	0/10	10/10	3000*	87.32	3000*	74.37	98.82	0.00	12.31	0.00
	2000	0/0	10/10	1/10	10/10	3000*	69.65	2704.66	771.47	98.98	0.00	11.19	0.00

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

Table 2.3: Results related to the optimality

$\|N\|$	Rd	# of UAVs				$Gap(\%)$		
		ES	EBP	PCS	PCBP	ES	PCS	PCBP
10	10	7.4	7.4	7.9	7.9	0.00	7.26	7.26
	20	4.1	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	18.4	11.1	12.2	12.2	67.20	10.15	10.15
	20	5.5	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 2.4: 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**

Table 2.5: 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.

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 2.6 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 2.7. The columns in Tables 2.6 and 2.7 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
- #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 2.3.1, the continuous relaxation neutralized the coverage constraints (2.4) and (2.20) 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 relaxation 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 2.6. 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 2.7 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 (2.21) 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 2.8 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 meters to 1,000 meters, #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 2.3 and 2.8 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.

Table 2.6: 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 2.7: 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 2.8: 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

2.5 Solutions and related problems of the USCP

In this section, the solution structure and the application of the USCP is presented. The optimal solution of the USCP consists of the set of demand points assigned to a UAV and its flight position. Because the UAVs have a predefined homogeneous network coverage radius, the optimal flight position of each UAV becomes a circular area, which has its size defined by the assigned demand points. In other words, the circumcircle of the chosen set of demand points can be smaller than the coverage radius, so there are multiple flight positions that are equivalently optimal. With a given network coverage radius R, if the set of demand points has a circumcircle with a circumradius R_c, then the optimal flight positions form a circle which has a radius $R - R_c$. For the sake of convenience, we use the term "optimal circle" to denote the circle of the optimal flight positions.

When the USCP is employed to the real application, a further decision would be required to select a flight position among many which are all "optimal" in the USCP. The authority can arbitrarily choose a flight position in the optimal circle, but also can fly in a *spotlight-mode*, which circulate along the boundary of the optimal circle, to consider the possible uncertainty of the location. The spotlight-mode flight along the optimal circle can maintain the connectivity to the selected demand points while covering the largest area. Therefore, it can increase the probability of detecting new casualties and the responsiveness to the possible movement of the current demand points.

In the UAV routing models mentioned in Section 2.1, the optimal circles of the USCP can be modeled as the clusters for the generalized traveling salesperson problem. A two-stage approach that identifies the candidate flight positions at the

first stage and plan the flight trajectory at the second stage can be used. The USCP can be used for the first stage. Under the given waypoints or the clusters, there is a vast literature considering the sequences (macroscopic-level planning) or the actual flight trajectory (microscopic-level planning) that can be used for the second stage.

Another related problem is the connected set covering problem (CSCP). As introduced in Section 2.2, when constructing the ad-hoc network by multiple UAVs, the UAVs should be close enough to communicate with each other and ultimately connected to the authority. The sets chosen for the solution should be *connected*, which forms a connected set covering problem. Existing literature on the ad-hoc sensor network models the connected set covering problem [121, 117, 51, 111, 58]. Because the connected set covering problem is NP-hard even for a graph with only one vertex with degree greater than two [99], existing research focuses on the approximation algorithm.

As the classical set covering problem, the existing literature of CSCP assumes the predefined candidate location of the facility. Thus, as the USCP can provide a more flexible and efficient solution than the classical set covering approach, the extra decision on the candidate location can provide a better solution than the classical CSCP. The CSCP is a special case of a group Steiner tree (GST) problem [42]. Garg et al. [46] formulated the GST as a MILP model. Wu et al. [111], and Huang et al. [58] referred the MILP model of Garg et al. [46] to approach minimum connected sensor covering problem and introduced approximation algorithms. Based on the columns generated during the USCP, the USCP can be developed into the connected set covering approach. Let Ω be the set of every column generated during the USCP. We assume that the flight position $(c^{\hat{x}_k}, c^{\hat{y}_k})$ of each UAV k to be the circumcenter

of the chosen set of demand points. For each pair of columns k and k' in Ω, the connectivity between two UAVs can be identified based on the distance between the flight positions. Let an access point of the authority be a root vertex, k_0. Also, each UAV (column) corresponds to a vertex. An undirected graph $G = (V, E)$ is defined, where $V = \Omega \cup \{k_0\}$ is the set of vertices and the set of the edge E is defined based on the binary connectivity between every two UAVs. To model the problem in GST, *group* g_i is defined for each demand point i. Group g_i consists of every vertex that covers the demand point i. As in the USCP, the decision variable z_k identifies the adoption of a vertex (or column, or UAV) k. A new continuous decision variable $\epsilon^i_{kk'}$ denotes the flow of the commodity i from vertex k to vertex k'. The following model represents the GST formulation of the problem:

$$\min \quad \sum_{k \in V} z_k \tag{2.22}$$

$$\text{s.t.} \quad \sum_{k \in V} w_{ik} z_k \geq 1, \qquad \forall i \in N \tag{2.23}$$

$$\sum_{(k,k') \in E} \epsilon^i_{kk'} - \sum_{(k,k') \in E} \epsilon^i_{k'k} = 0, \qquad \forall i \in N, \forall k \in V \setminus (\{k_0\} \cup g_i) \tag{2.24}$$

$$\sum_{(k_0,k) \in E} \epsilon^i_{k_0 k} = \sum_{k \in g_i} \Big(\sum_{(k,k') \in E} \epsilon^i_{k'k} - \sum_{(k,k') \in E} \epsilon^i_{k'k} \Big), \quad \forall i \in N \tag{2.25}$$

$$\epsilon^i_{kk'} \leq z_k, \qquad \forall i \in N, \forall k, k' \in V \tag{2.26}$$

$$\epsilon^i_{kk'} \leq z_{k'}, \qquad \forall i \in N, \forall k, k' \in V \tag{2.27}$$

$$z_k \in \mathbb{B}, \qquad \forall k \in V \tag{2.28}$$

$$\epsilon^i_{kk'} \in [0,1] \qquad \forall i \in N, \forall k, k' \in V \quad (2.29)$$

In the model, the network flow assures the connection of the UAVs. As in the USCP, every demand point should be covered. By Constraints (2.24) and (2.25), at least one flow should stream from the root vertex to at least one vertex in every group. The network flow and the adoption of a UAV is related by Constraints (2.26) and (2.27).

The existing approaches cannot be used for the CSCP when the network structure is not given. However, in the USCP-based approach introduced above, the network structure is defined while executing the CG for the USCP and can be used to solve the CSCP. If the optimal solution for the USCP is found, but the GST problem comes out to be infeasible, it implies that the current G is not a connected graph. In this case, one can generate candidate UAVs (or vertices) between the individual connected subgraphs by heuristics to ensure the graph to be connected.

2.6 Summary

This chapter 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. To use Dantzig-Wolfe decomposition over the quadratic coverage distance

constraint, the generic framework of Vanderbeck and Savelsbergh [106] is used. The finite generating set is designed using the projection of the feasible set on the discrete decision variables. An extended formulation and the associated B&P algorithm, which were developed for the stronger relaxation 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. One can achieve a more optimal solution by the consideration of the extra decision of location. A similar approach can be used in various problems, and the solution approaches of the USCP can be implemented while solving them.

Chapter 3

Unmanned aerial vehicle variable radius set covering problem

3.1 Introduction

In Chapter 2, the location problem of UAVs were modeled as a set covering problem with a fixed-radius coverage constraint. The main advantage of the UAV is its flexible, unrestricted positioning, which also complicates the decision of the operation planning. In the unmanned aerial vehicle set covering problem considering fixed-radius coverage constraint (USCP), the UAVs were assumed to fly at a certain altitude while providing the wireless network on the circular area with a fixed coverage radius. However, in realistic applications, UAV positions at various altitudes and their corresponding coverage distance can be diversified. Literature considers the relation between the UAV altitude, the coverage radius, and the operation cost [38, 122].

The concept of the coverage distance has been investigated by numerous researchers, as introduced in Chapter 2. As mentioned in Berman et al. [14], the radius of the coverage area can vary based on the capability and the height of the facility. Also, it is natural that the coverage size of each facility follows a *monotonically increasing function* of its cost, and vise versa. The proposed problem in

Berman et al. [14] consists of the characteristic of the UAV operation problem even before the UAVs took the limelight. As told in the previous chapters, the continuous facility location problem introduced by Drezner et al. [40] and Plasteria [85] tries to locate one facility without candidate positions. Berman et al. [14] named the continuous, multi-facility location problem on the xy-plane as a "planar version" of the set covering problem.

The planar version of the variable radius covering problem is closely related to the set covering approach of the UAV operation problem. Not only for the naturalness of the UAV altitude-coverage radius relation but also the altitude-cost relation are similar to the variable radius covering model. When UAVs are being operated, the thrust and performance of their propellers decreases in higher altitudes [73]. The hovering and maneuvering energy consumption rate increases in higher altitudes because of the lower air density [82]. Also, the maximum flight speed on the vertical flight is significantly slower than on the horizontal flight and requires substantial energy consumption, too. Thus, the cost of the flight to higher altitudes increases not only because of the longer flight distance but also because of inherent difficulty. The monotonic increase of the cost for larger coverage distances is a realistic assumption, even in the case of UAV operation. The cost function of the sum of the fixed cost per facility and the quadratically increasing cost related to the coverage distance is also valid and used in this research.

The *planar version* of the covering problem is considered in this research. However, the term "planar" represents one aspect of the problem among many, so the term "USCP" and "Euclidean formation" defined in Chapter 2 is used throughout this chapter, instead. In the USCP, we modeled the UAV location problem with a

covering-based approach. The coverage radius of the homogeneous UAV is given as a parameter, and the UAV can locate on the xy-plane without any restriction. The objective of the problem is to cover every demand point with a minimum number of UAVs. In this research, we *extended* the USCP by allowing the decision on the coverage radius of the UAV. The term "extended" is used because the problem considers an extra decision from the USCP, and the solution algorithms designed for the USCP cannot solve the problem in this research. The objective is minimizing the total cost of the UAV operation, which is the sum of the fixed cost related to the number of the UAVs and the operation cost related to the coverage radius of each UAV. For the unity of logic, we relate the problem to the USCP and rename it as UAV variable radius set covering problem (UVCP).

As the set covering approach of the facility location problem discretizes the decision and optimizes it over the given candidate locations, the ordinary UAV operation problems, such as set covering and routing, limit the location decision on the discrete levels of altitudes [38, 122]. The model proposed in this research decides the coverage radius over the continuous domain, which signifies the continuous decision of the UAV altitude. This approach can improve the solutions of the aforementioned research and the possible applications introduced in Chapter 2, which will provide more efficient operations in realistic applications.

Even though Berman et al. [14] introduced the variable radius set covering problem, they claimed that they could not provide a mathematical formulation explicitly stated to be solved by mathematical programming solvers. In this research, we modeled the UVCP in the mixed-integer quadratically constrained quadratic program with the techniques proposed in Chapter 2 and the USCP. By the Dantzig-Wolfe

decomposition, the UVCP is reformulated to the extended formulation. From each UAV point of view, the demand points covered by a UAV are treated in a bundle, which allows the branch-and-price algorithm to solve the exact problem efficiently. In particular, we could use the "minimal subset" concept which Berman et al. [14] proposed but could not use in their solution algorithm. To find every minimal subset of the UVCP, they had to use the brute force procedure, checking the exponential number of subsets. In this research, we could consider only requiring a minimal subset by using the column generation algorithm.

However, we could not use the techniques of the USCP as they stood. In the UVCP, the coverage radius is modeled as a continuous decision variable, so the continuous relaxation of the problem provides a non-convex feasible set, which makes solving the problem with a commercial solver impossible. We proposed a solvable equivalent subproblem that consisted of the linear constraints by the Heron's formula and the minimal covering circle. The equivalent subproblem required a larger number of constraints and could be challenging to solve with a large-sized problem. Thus, we proposed in this paper a heuristic that discretizes and fixes the coverage radius and solves with the USCP. The proposed algorithms outperformed the genetic algorithm (GA) of Berman et al. [14] in the computational experiments. The remainder of this chapter is organized as follows: Section 3.2 defines the problem and proposes the mathematical model. Section 3.3 presents the reformulation of the UVCP and the corresponding branch-and-price algorithm. Section 3.4 proposes the solvable equivalent subproblem and describes its equivalence. The fixed coverage radius-based heuristic is introduced in Section 3.5. The proposed algorithms are compared with the computational experiments in Section 3.6. Finally, the chapter

is summarized in Section 3.7.

3.2 Problem definition

This section presents a detailed description of the UVCP. As in the USCP, the UAVs construct wireless networks to cover demand points. The set covering approach is used, so the battery constraint and the flight to each location are not considered because of the simplification. The problem focuses on covering every demand point by minimizing the use of resources. As was assumed in the USCP, the problem, too, is designed for response planning in disaster situations. Therefore, it needs to be robust enough to respond to every possible demand. Even though the objective is to minimize the total use of resources, this is not done for profit maximization in emergency situations. As the USCP minimizes the cardinality of UAVs, efficient operation is essential to securing additional responsiveness. By minimizing total resource usage, authorities can ensure maximum responsiveness in rapidly changing environments. The assumptions of the presented problem are defined as follows, which has only one difference from the USCP:

(1) The information on the positions of demand points is already known.

(2) There is no restriction on a UAV's hovering position in the xy-plane.

(3) A demand point is covered if it is in the coverage circle.

(4) There is no transmission capacity limitation on the wireless network.

(5) There is no overlap interference between UAVs or shadowing effects incurred by buildings.

(6) UAVs have different coverage distances.

Only the last assumption is different from the USCP. Every UAV can fly at any altitude, which provides the various coverage distance of the wireless network. The minimum and maximum limit of the coverage distance is not considered, yet those constraints are easy to consider in the solution algorithm. The existing work of Berman et al. [14] did not consider the limitation of the coverage radius because it was somewhat challenging to implement in the GA. In order to compare the performance of the solution algorithm, they are not considered in the UVCP. However, those limitations do not complicate the solution algorithm's structure and are rather advantageous from the standpoint of computation speed.

As assumed in the USCP, the information of the demand points is given as primary data and can be updated later with the operation of the UAVs. The UAVs' resource consumption is defined as a summation of the fixed cost and the coverage radius-related cost. It is natural to minimize the number of UAVs operated, as introduced in the USCP. Furthermore, it is also important to minimize the flight altitude of each UAV and the corresponding coverage radius in order to maintain the flight capability of the UAVs.

Figure 3.1 (below) presents an overview of the UVCP. UAVs can be located freely on the xy-plane, so they are not bounded in the predefined candidate positions. The coverage radius of the wireless network can be decided without a restriction. Note that the network areas in Figure 3.1 provide tight circles around the demand points assigned to the UAVs. Compared to Figure 2.1, which assumes the fixed-radius coverage, the UVCP tries to minimize the resource usage, so the networks form a tight circle while covering every demand assigned to each UAV. This relates to the

concept of a minimal subset, which is introduced later.

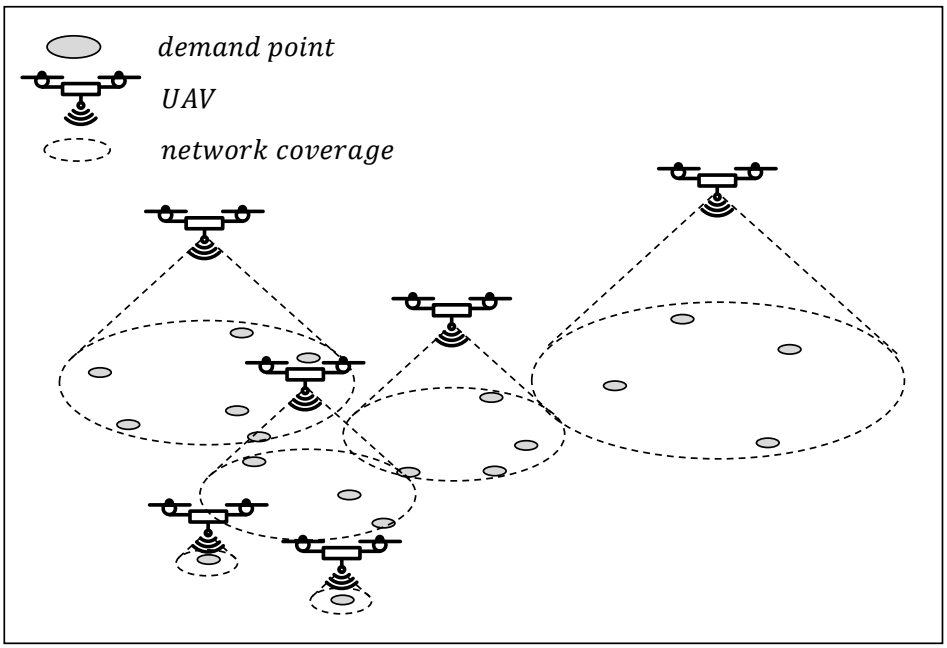


Figure 3.1: Overview of the UVCP

3.2.1 Mathematical model

Berman et al. [14] proposed a mathematical model of the UVCP and named it as the planar variable radius covering problem. For the unity of the research, the notations are modified as follows:

Set

N set of demand points.

Parameters

$a_i^{\hat{x}}$ position of demand point i on x-coordinate. $\forall i \in N$

$a_i^{\hat{y}}$ position of demand point i on y-coordinate. $\forall i \in N$

F fixed cost of operating one UAV.

Decision variables

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

$c_j^{\hat{x}} \in \mathbb{R}$, position of UAV j on x-coordinate. $\forall j \in \{1, ..., |N|\}$

$c_j^{\hat{y}} \in \mathbb{R}$, position of UAV j on y-coordinate. $\forall j \in \{1, ..., |N|\}$

$r_j \in \mathbb{R}$, coverage radius of UAV j. $\forall j \in \{1, ..., |N|\}$

$$x_{ij} = \begin{cases} 1, & \text{if demand point } i \text{ is allocated to UAV } j. \\ 0, & \text{otherwise.} \end{cases} \qquad \begin{matrix} \forall i \in N, \\ \forall j \in \{1, ..., |N|\} \end{matrix}$$

Note that the decision variable y_j was not used in the original problem of [14]. Instead, they claimed that the number of the operating facilities (in this research, UAVs) should be a decision variable, p, and the index of the j should be an element of the set, $\{1, ..., p\}$. The self-dependence of the index and the decision variable precluded the usage of the mathematical programming solver. However, as stated in Chapter 2 and the USCP, the number of UAVs operated in the UAV covering problem does not exceed the number of demand points, $|N|$. We propose a binary decision variable y_j which indicates the operation of the UAV j, and use the set, $\{1, ..., |N|\}$. Berman et al. [14] defined $d_i\big((c_j^{\hat{x}}, c_j^{\hat{y}})\big)$ as the Euclidean distance between demand point i and UAV j, and the corresponding cost, $\phi(r)$, as a variable operation cost of a UAV of the coverage radius r. They referred to Drezner [39] and Fernández

et al. [45] to define a proper cost structure related to the coverage distance, and chose the quadratic relation, $\phi(r) = r^2$. The original formulation proposed by Berman et al. [14] is presented as follows:

[Original formulation of Berman et al. [14]]

$$\min \quad pF + \sum_{j=1}^{p} \phi\left(\max_{i \in N} \left\{ x_{ij} d_i\big((c_j^{\hat{x}}, c_j^{\hat{y}})\big) \right\} \right), \tag{3.1}$$

$$\text{s.t.} \quad \sum_{j=1}^{p} x_{ij} \geq 1, \qquad \forall i \in N \tag{3.2}$$

$$x_{ij} \in \mathbb{B}, \qquad \forall i \in N, j \in \{1, ..., p\} \tag{3.3}$$

Berman et al. [14] claimed that the UVCP is not suitable for a mathematical optimization approach because of the self-dependence of the index and the inexplicit expression of the coverage radius, $d_i\big((c_j^{\hat{x}}, c_j^{\hat{y}})\big)$. However, based on the approaches proposed in Chapter 2, an explicit formulation is designed in the form of the mixed-integer quadratically constrained quadratic programming.

[Explicit Reformulation of the UVCP]

$$\min \quad \sum_{j=1}^{|N|} y_j F + \sum_{j=1}^{|N|} \phi(r_j), \tag{3.1$'$}$$

$$\text{s.t.} \quad x_{ij} \leq y_j, \qquad \forall i \in N, \forall j \in \{1, ..., |N|\} \tag{3.4}$$

$$\sum_{j=1}^{|N|} x_{ij} \geq 1, \qquad \forall i \in N \tag{3.2$'$}$$

$$(a_i^{\hat{x}} - c_j^{\hat{x}})^2 + (a_i^{\hat{y}} - c_j^{\hat{y}})^2 \leq r_j^2 + \tilde{M}(1 - x_{ij}), \quad \forall i \in N, \forall j \in \{1, ..., |N|\} \tag{3.5}$$

$$x_{ij} \in \mathbb{B}, \qquad \forall i \in N, \forall j \in \{1, ..., |N|\} \tag{3.3$'$}$$

$$y_j \in \mathbb{B}, \qquad \forall j \in \{1, ..., |N|\} \tag{3.6}$$

$$c_j^{\hat{x}}, c_j^{\hat{y}}, r_j \in \mathbb{R} \qquad \forall j \in \{1, ..., |N|\} \tag{3.7}$$

The explicit reformulation of UVCP looks similar to the Euclidean standard (ES) formulation of the USCP except for two different points. First, the objective function is different. As modeled in Berman et al. [14], we assume the variable operation cost, $\phi(r)$, to be r^2. Thus, the problem becomes quadratic programming. Second, the coverage distance is modeled as a parameter in the USCP but is a continuous decision variable in the UVCP.

Even though the problem is modeled in an explicit mathematical formulation, because of Constraint (3.5), the feasible set of the continuous relaxation of the explicit reformulation is not convex. Therefore, unlike the ES formulation, the explicit reformulation cannot be solved by the commercial solvers, which utilize the linear programming (LP) relaxation-based branch-and-cut algorithms for the optimization. Furthermore, in the USCP, the ES formulation could not even solve the smallest instances within an applicable computation time because of the weak continuous relaxation bounds and the symmetries of the problem. Considering that the UVCP has an extra decision on the coverage radius r, it would be extremely difficult to solve the explicit reformulation in its current form. To solve the problem, first, we decomposed the explicit reformulation of the UVCP to the extended formulation. As in the USCP, we used the fact that it is easy to find the optimal position and the size of the minimum covering circle if the set of demand points is given. By using similar generating sets as in the USCP, the UVCP is decomposed based on the decision variable x_{ij}. In the explicit reformulation of the UVCP, it is only necessary

to show that Constraint (3.5) can be formulated linearly. In Section 3.4, the pricing subproblem II consists of mixed-integer linear constraints and has a polyhedral feasible set. Section 3.4.2 proves that two pricing subproblems are equivalent so that the UVCP can be solved by the branch-and-price (B&P) algorithm.

3.3 Branch-and-price approach to the UVCP

We reformulate the problem into the set-partitioning extended formulation. Column-wise decisions on demand points that are covered by a UAV are modeled by the Dantzig-Wolfe decomposition. Because a set of demand points can define a unique minimum covering circle, Berman et al. [14] defined this as a *minimal subset*, which consists of x_{ij} and the corresponding r_j and $(c_j^{\hat{x}}, c_j^{\hat{y}})$. Each minimal subset is fully described by the decision of the set of the demand points, x_{ij}. Berman et al. [14] showed that there is an optimal solution of the UVCP that consists of the minimal subsets. While using the Dantzig-Wolfe decomposition, the decision on the set of the demand points can describe a column to consist of an extended formulation. When the set of demand points is chosen, a unique minimum covering circle is defined. Because the coverage radius, r_j, equals the radius of the minimum covering circle, the cost of the column, g_k, is defined as $F + \phi(r_k)$. Let Ω be the set of every column that defines a feasible set of demand points covered by one UAV. A binary decision variable z_k identifies the usage of column k, and an allocation of demand point i to column k is described by a binary parameter w_{ik}. The extended formulation of the UVCP is defined as follows:

[Extended formulation]

$$\min \quad \sum_{k \in \Omega} g_k z_k, \tag{3.8}$$

$$\text{s.t.} \quad \sum_{k \in \Omega} w_{ik} z_k \geq 1 \qquad \forall i \in N \tag{3.9}$$

$$z_k \in \mathbb{B} \qquad \forall k \in \Omega \tag{3.10}$$

The extended formulation of the UVCP consists of the same constraints as the USCP and a slightly different objective function (3.8), which minimizes the total resources required to operate UAVs to cover every demand point. Constraint (3.9) forces the problem to cover every demand point at least once. Berman et al. [14] showed that there exist, at most, $|N|(|N|^2 + 5)$ minimal subsets. Even though there are $2^n - 1$ possible subsets, which means $2^n - 1$ feasible columns defining set Ω, only a part of the subset is required for the extended formulation to describe the original UVCP. However, Berman et al. [14] did not have a mathematical tool to find the required minimal subsets and claimed that one should use the brute force search. Instead of enumerating every possible column, which is exponential, the column generation (CG) algorithm can provide new columns, which might improve the solution from the current set of columns. The pricing subproblem only provides solutions that are minimal subsets and will likely produce only a minimum number of columns to describe the solution set of the UVCP fully. Is introduced in the previous chapters, in the CG algorithm, the LP relaxation of the extended formulation with the current columns is called the *restricted master problem* (RMP). Let π_i be a dual price of Constraint (3.9) in the RMP. By solving the pricing subproblem, one can find

a set of demand points with a minimum reduced cost. If the objective value provides a negative reduced cost, the new column is generated and included in the set Ω. The pricing subproblem I has the same decision variables defined in the Euclidean branch-and-price (EBP), except a coverage radius r.

[Pricing subproblem I]

$$\min \quad F + \phi(r) - \sum_{i \in N} \pi_i x_i \tag{3.11}$$

$$\text{s.t.} \quad (a_i^{\hat{x}} - c^{\hat{x}})^2 + (a_i^{\hat{y}} - c^{\hat{y}})^2 \leq r^2 + \tilde{M}(1 - x_i), \qquad \forall i \in N \tag{3.12}$$

$$x_i \in \mathbb{B}, \qquad \forall i \in N \tag{3.13}$$

$$r \in \mathbb{R}, \tag{3.14}$$

$$c^{\hat{x}}, c^{\hat{y}} \in \mathbb{R} \tag{3.15}$$

Because the dual variable π_i has a nonnegative value, the pricing subproblem I, which is a minimization problem, only provides a solution as circle-defining demand points with every demand point included in the circle. Thus, the solution of the pricing subproblem I is always a minimal subset. The subproblem I has an intuitive structure. Particularly, it is known that in the case of the fixed radius–the USCP–the subproblem of the EBP is solved efficiently by the commercial solver. However, as mentioned in Section 3.2.1, the continuos relaxation of the feasible set is nonconvex because of Constraint (3.12). Therefore, the optimization solver cannot find the optimal solution of the pricing subproblem I. Instead, an equivalent pricing subproblem can be defined based on the knowledge of Heron's formula and the minimum covering circle.

3.4 Minimum covering circle-based approach

3.4.1 Formulation of the pricing subproblem II

The coverage constraint of the pricing subproblem I is modeled in a mixed-integer quadratic formulation. Constraint (3.12) relates a circle and a demand point by the distance between the center of the circle and the demand point and the radius of the circle. However, a circle can also be defined by two or three points. A minimum covering circle of two points is a circle whose diameter is the line connecting two points. When covering three points, if three points form an obtuse triangle, two points on the obtuse angle can define a minimum covering circle. If three points form a right or an acute triangle, a circumcircle defined by three points is a minimum covering circle. When a set of demand points is fixed, it is obvious that a minimum covering circle of the set has the largest radius among the minimum covering circles of its subsets. Thus, the constraint is valid for every two-element and three-element subset. On the other hand, because there exists a two-element or three-element subset that defines the minimum covering circle, by modeling the constraint for every two-element and three-element subset, the coverage distance constraint can be defined.

To model the minimum covering circle-based formulation, we define a parameter $d_{ab} := ||a-b||_2$ as a distance between two demand points. Also, the distance between a demand point i and the UAV j, d_{ij}, can be defined as a decision variable instead of as the distance function, $d_i\big((c_j^{\hat{x}}, c_j^{\hat{y}})\big)$, used in Berman et al. [14]. However, if a position of a UAV j, $(c_j^{\hat{x}}, c_j^{\hat{y}})$, is given, d_{ij} can be calculated as being in between two demand points. For demand points a, b, and c, without loss of generality, let

$d_{ab} \geq d_{bc}$ and $d_{ab} \geq d_{ca}$. Based on what we know of the minimum covering circle, a radius of the covering circle, R_{abc}, is defined as:

$$R_{abc} := \begin{cases} d_{ab}/2, & \text{if } d_{bc}^2 + d_{ca}^2 \leq d_{ab}^2 \\ \dfrac{d_{ab}d_{bc}d_{ca}}{4\sqrt{S_{abc}(S_{abc}-d_{ca})(S_{abc}-d_{bc})(S_{abc}-d_{ab})}}, & \text{otherwise,} \end{cases}$$

where $S_{abc} := (d_{ab} + d_{bc} + d_{ca})/2$.

The pricing subproblem II is defined based on the same decision variables $\boldsymbol{x}$ and r of pricing subproblem I, but without $c^{\hat{x}}$ and $c^{\hat{y}}$.

[Pricing subproblem II]

$$\begin{aligned} \min \quad & (3.11) \\ \text{s.t.} \quad & r \geq R_{i_1 i_2 i_3} + \tilde{M}'(x_{i_1} + x_{i_2} + x_{i_3} - 3), && \forall i_1, i_2, i_3 \in N && (3.16) \\ & r \geq d_{i_1 i_2}/2 + \tilde{M}'(x_{i_1} + x_{i_2} - 2), && \forall i_1, i_2 \in N && (3.17) \\ & (3.13), (3.14) \end{aligned}$$

The pricing subproblem II has the same objective function as the pricing subproblem I. As mentioned above, the constraint relating to the minimum coverage radius and the set of demand points can be defined by the constraint over every two-element and three-element subset. The radius of the minimum covering circle, $R_{i_1 i_2 i_3}$, is defined for every three-element subset of I. Constraint (3.17) ensures that the minimum coverage radius should be the same or larger than the radius of the minimum covering circle of any three demand points chosen in the solution of the subproblem. Constraint (3.16) relates the minimum coverage radius to the radius of

the minimum covering circle defined by two demand points.

In the subproblem II, the coverage distance is modeled by the intersection of the multiple mixed-integer linear constraints instead of by the mixed-integer quadratic constraint. Most importantly, the subproblem II has a polyhedral feasible set, and the LP relaxation of the subproblem is convex. It implies that the UVCP has a polyhedral feasible set as well and can be modeled in Minkowski representation. Accordingly, the UVCP can be solved by the B&P algorithm, and at the same time, the subproblem can be solved by the optimization solvers. One coverage constraint of subproblem II corresponds to one minimal subset defined in Berman et al. [14]. As claimed in Berman et al. [14], there exists an optimal solution of the UVCP that is a union of the minimal subset, and each minimal subset can be defined from a solution of the pricing subproblem II. Berman et al. [14] could not model the concept of the minimal subset in a mathematical model and claimed that every instance of the power set should be checked by the brute force approach. Instead of the exponential enumeration, they proposed heuristic approaches, including GA.

However, the subproblem II requires a larger number of constraints to define the same problem than does the subproblem I. The subproblem I requires $|N|$ constraints to define the coverage distance, whereas the subproblem II requires $|N|(|N|^2+5)/6$ constraints. The same technique can be applied to the reformulation of the standard problem, but there are two limitations. As analyzed in Chapter 2, the continuous relaxation neutralizes the coverage distance constraint and makes meaningless value of the continuous relaxation bound. The only difference is that in the ES in the USCP, the objective is to minimize the number of UAVs, and the continuous relaxation bound is 1, whereas in the UVCP, the continuous relaxation bound equals the

fixed cost, F.

Another limitation is the size of the problem. Because it requires $|N|^2(|N|^2+5)/6$ constraints to define coverage distance constraints, the size of the problem increases quickly as the number of the demand points increases. For example, if $|N|$ is equal to 50 and 100, then 1,043,750 and 16,675,000 constraints are required, respectively. It was observed that the problems of more than 50 nodes could not even be loaded to the commercial solver. Even for problems with only ten nodes, the commercial solvers could not solve the problem within one hour because of the weak lower bounds and the symmetries of the branching tree.

3.4.2 Equivalence of the subproblem

The equivalence of two subproblems is analyzed in this section. Although the feasible regions of the continuous relaxation of two subproblems are different, the integer decision variable bounds the feasible region, making them equivalent.

Theorem 3.1. *Pricing subproblems I and II are equivalent.*

Because the two problems have the same objective function, it is only necessary to show that the feasible set of both problems is the same. Let a feasible set of the pricing subproblem I be S_I and a feasible set of the pricing subproblem II be S_{II}.

Proof. ($\Rightarrow$) We show that $S_I \subseteq S_{II}$. Let a feasible solution $\psi = (c^{\hat{x}}, c^{\hat{y}}, r, \boldsymbol{x}) \in S_I$. Using the proof of contradiction, let us assume that $\exists\, i_1, i_2, i_3 \in J$ such that $r, x_{i_1}, x_{i_2}, x_{i_3} \notin S_{II}$. We are focused on the case of $i_1 \neq i_2 \neq i_3$, because if there are identical points, it is self-evident that $x_{i_1}, x_{i_2}, x_{i_3} \in S_{II}$. There are two cases related to the distances between the demand points.

(Case 1) $d^2_{i_1 i_2} \geq d^2_{i_2 i_3} + d^2_{i_3 i_1}$ (Demand points are on the same straight line or form an obtuse triangle). By the assumption, the following holds:

$$r + \tilde{M}'(2 - x_{i_1} - x_{i_2}) < \frac{d_{i_1 i_2}}{2}.$$

Because $x_{i_1}, x_{i_2} \in \mathbb{B}$ and $r \geq 0$, $x_{i_1} = x_{i_2} = 1$ and $r \leq \frac{d_{i_1 i_2}}{2}$.

$d_{i_1 j} \leq r$, $d_{i_2 j} \leq r\,(\because \psi \in S_I) \implies d_{i_1 j} + d_{i_2 j} \leq 2r < d_{i_1 i_2}$, which is false by the triangle inequality.

(Case 2) $d^2_{i_1 i_2} < d^2_{i_2 i_3} + d^2_{i_3 i_1}$. Similar to Case 1, the following holds:

$$r + \tilde{M}'(3 - x_{i_1} - x_{i_2} - x_{i_3}) < R_{i_1 i_2 i_3} \implies x_{i_1} = x_{i_2} = x_{i_3} = 1, r < R_{i_1 i_2 i_3}.$$

However, for i_1, i_2 satisfying $d_{i_1 j} \leq r$, $d_{i_2 j} \leq r$, $d_{i_3 j} > R_{i_1 i_2 i_3}$ holds, which proves the false assumption.

($\Leftarrow$) We show that $S_{II} \subseteq S_I$. Let a feasible solution $\nu = (r, \boldsymbol{x}) \in S_{II}$. For set of demand points that satisfies $x_i = 1$ in a solution ν, there exists a unique minimum covering circle, and the circle can be defined by at most three points in the set.

If the circle is defined by two points, i_1 and i_2, $(c^{\hat{x}}, c^{\hat{y}}, r, \boldsymbol{x})$ satisfies Constraint (3.12), where the position $(c^{\hat{x}}, c^{\hat{y}})$ is defined as:

$$c^{\hat{x}} = \frac{a^{\hat{x}}_{i_1} + a^{\hat{x}}_{i_2}}{2}, \quad c^{\hat{y}} = \frac{a^{\hat{y}}_{i_1} + a^{\hat{y}}_{i_2}}{2}.$$

If the circle is defined by three points, i_1, i_2, and i_3, the solution is defined as:

$$c^{\hat{x}} = \frac{((a^{\hat{x}}_{i_1})^2 + (a^{\hat{y}}_{i_1})^2)(a^{\hat{y}}_{i_2} - a^{\hat{y}}_{i_3}) + ((a^{\hat{x}}_{i_2})^2 + (a^{\hat{y}}_{i_2})^2)(a^{\hat{y}}_{i_3} - a^{\hat{y}}_{i_1}) + ((a^{\hat{x}}_{i_3})^2 + (a^{\hat{y}}_{i_3})^2)(a^{\hat{y}}_{i_1} - a^{\hat{y}}_{i_2})}{2[a^{\hat{x}}_{i_1}(a^{\hat{y}}_{i_2} - a^{\hat{y}}_{i_3}) + a^{\hat{x}}_{i_2}(a^{\hat{y}}_{i_3} - a^{\hat{y}}_{i_1}) + a^{\hat{x}}_{i_3}(a^{\hat{y}}_{i_1} - a^{\hat{y}}_{i_2})]}$$

$$c^{\hat{y}} = \frac{((a^{\hat{x}}_{i_1})^2 + (a^{\hat{y}}_{i_1})^2)(a^{\hat{x}}_{i_3} - a^{\hat{x}}_{i_2}) + ((a^{\hat{x}}_{i_2})^2 + (a^{\hat{y}}_{i_2})^2)(a^{\hat{x}}_{i_1} - a^{\hat{x}}_{i_3}) + ((a^{\hat{x}}_{i_3})^2 + (a^{\hat{y}}_{i_3})^2)(a^{\hat{x}}_{i_2} - a^{\hat{x}}_{i_1})}{2[a^{\hat{x}}_{i_1}(a^{\hat{y}}_{i_2} - a^{\hat{y}}_{i_3}) + a^{\hat{x}}_{i_2}(a^{\hat{y}}_{i_3} - a^{\hat{y}}_{i_1}) + a^{\hat{x}}_{i_3}(a^{\hat{y}}_{i_1} - a^{\hat{y}}_{i_2})]}.$$

Both $S_I \subseteq S_{II}$ and $S_{II} \subseteq S_I$ are shown and proves the claim. $\square$

One can solve the CG of the UVCP with the pricing subproblem II by Theorem 3.1.

3.5 Fixed-radius heuristic

The pricing subproblem II has an LP relaxation with a convex feasible set. Also, the subproblem consists of linear constraints so that the commercial solver can handle the problem in a more numerically stable procedure. Thus, the corresponding B&P algorithm is an exact algorithm that can solve the problem in a short computation time. However, as mentioned above, the pricing subproblem II required $|N|(|N|^2+5)/6$ constraints to model the coverage distance constraint for N demand points. The size of constraints can rapidly increase in the case of a large number of demand points. On the other hand, the pricing subproblem I required one constraint per one demand point, so the size of the subproblem does not drastically increase uncontrollably. In a similar vein, the formulation of the subproblem II can be utilized to solve the USCP if the fixed coverage radius is used instead of the variable coverage radius. However, this formulation is developed to overcome the nonconvexity of the UVCP, so it would be better to avoid the unnecessary risk of the larger number of constraints.

In order to utilize the findings of the USCP that the subproblem of the EBP could provide to the solution of the large-sized problem in a short computation time, we developed a heuristic. The pricing subproblem I is not solvable by itself. The fixed-radius heuristic (FRH) proposed in this research solved the subproblem I by fixing the coverage radius, which had the same feasible set of the subproblem of the EBP. In one iteration of the CG algorithm, the FRH investigated various radius r in a discrete manner and choose the radius that provided the most negative reduced cost

to generate a new column. To put it succinctly, the FRH solved the approximation of the original pricing subproblem I by the decomposition over the various radius r. The continuous relaxation of each decomposed problem has the convex feasible set, and can be solved by the optimization efficiently. The mathematical formulation of the decomposed problem of radius R is provided as follows:

$$\begin{aligned} \min \quad & (3.11) \\ \text{s.t.} \quad & (2.17) - (2.19) \end{aligned}$$

The solution of the decomposed problem includes the set of demand points that is covered by one UAV. When the cost is realized to calculate the reduced cost and generate the column, the radius of the minimum covering circle of the chosen demand points is used instead of the originally fixed radius, which has the same or a smaller value. Let a set $\bar{N}$ be the demand points chosen in the solution of the decomposed problem. Welzl's algorithm [110] can be used to derive the radius of the minimum covering circle. Also, a modified optimization problem of the pricing subproblem I, which is related to the continuous single facility location problem, can be used:

$$\min \quad r \tag{3.18}$$

$$\text{s.t.} \quad (a_i^{\hat{x}} - c^{\hat{x}})^2 + (a_i^{\hat{y}} - c^{\hat{y}})^2 \leq r^2, \qquad \forall i \in \bar{N} \tag{3.19}$$

$$(3.14), (3.15)$$

Because the original problem of the FRH, pricing subproblem I, has the non-convex feasible set, the FRH or the hybrid algorithm of the FRH and the Newton's method cannot find every optimal solution of the original problem. However, as

mentioned above, the actual radius r is changed to be locally optimal after fixing the demand points to be covered, so it is tentatively expected that the heuristic can work as a sieve under the appropriate decomposition interval that can find the most of the effective solutions. In one iteration of the CG, the FRH can stop when a solution with the negative reduced cost is found or can check every possible radius r and generate a column with the most negative reduced cost. The procedure of the FRH is presented in Figure 3.2 (below).

Also, the FRH can be implemented in the hybrid algorithm to speed up the CG algorithm for large-sized problems. In the initial stage of the CG, the FRH can be used instead of the subproblem to provide decent columns rapidly, and when the FRH cannot find a negative reduced cost, pricing subproblem II is used for the CG to converge to the optimal. The performance of the FRH and the hybrid algorithm are compared in Section 3.6.

3.6 Computational experiments

The computational experiments were conducted to compare the performances of the proposed solution algorithms to the benchmark genetic algorithm proposed in Berman et al. [14]. The dataset used in the computational experiment is introduced in Section 3.6.1. The B&P algorithm and the FRH were developed in FICO Xpress 8.5 and solved with Xpress-Optimizer 33.01.02. Because the coverage distance-related cost function was modeled as a quadratic function, the pricing subproblem II was solved by the Xpress MIQP solver. The FRH was operated in Xpress MIQCQP solver using the barrier method. The genetic algorithm was developed and solved in Python 3.6. Experiments were performed with an AMD Ryzen ™ 7

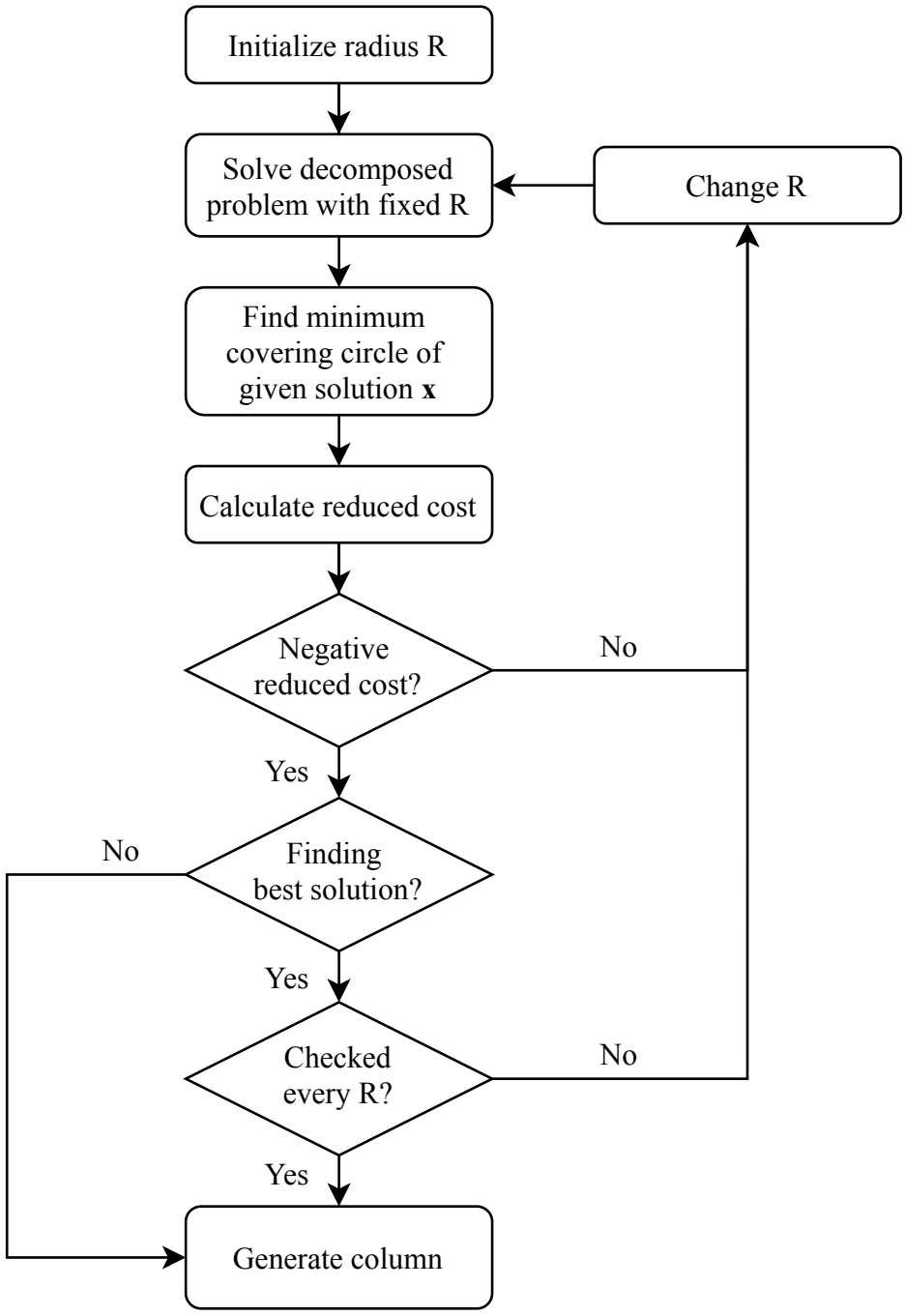


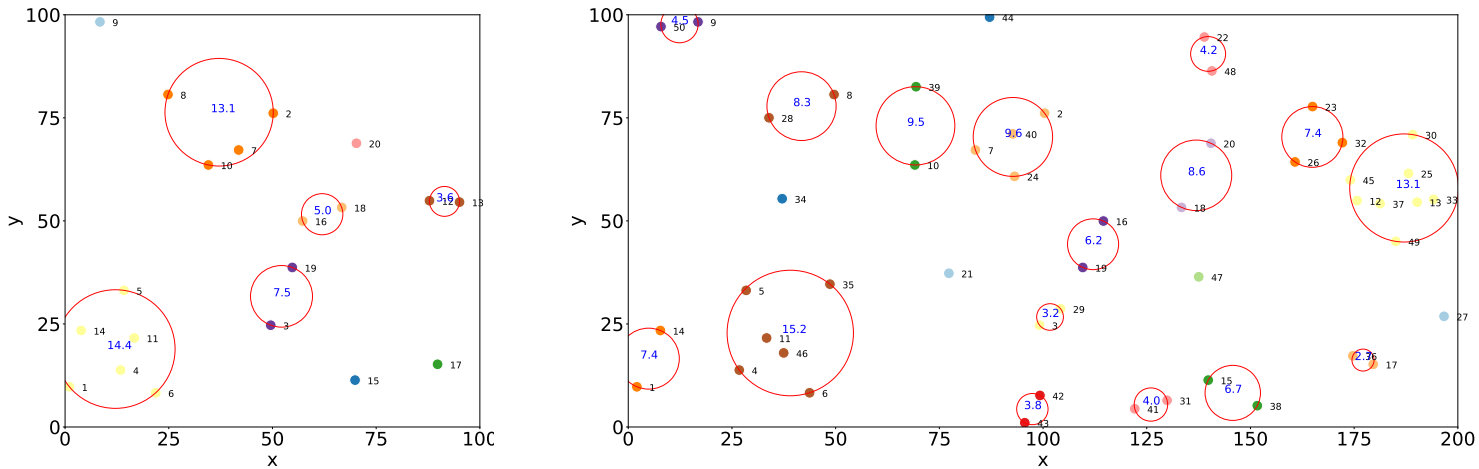
Figure 3.2: The procedure of the fixed-radius heuristic

2700X 8-Core CPU at 3.70GHz and 32GB of RAM running on a Windows 10 64-bit operating system.

3.6.1 Datasets used in the experiments

In the experiment, the fixed cost of a UAV, F, is set to be 100 and the variable operation cost function is set to be $\phi(r) = r^2$, following Berman et al. [14]. We experimented with the algorithms for the randomly generated problems, following Berman et al. [14]. Two types of problems were generated, whose uniformly distributed demand points were plotted on a $100\times$ 100 square and a 200×100 rectangle, respectively. Four sizes of demand points were tested: 10, 20, 50, and 100. In total, eight problem classes were generated, and ten instances were generated for each problem class. Even though Berman et al. [14] tested datasets with larger sizes of demand points, we decided to limit the size of the problem because the best feasible solutions found in Berman et al. [14] were not realistic, and we observed that even in the problems we tested in this research, the GA did not converge to the optimal.

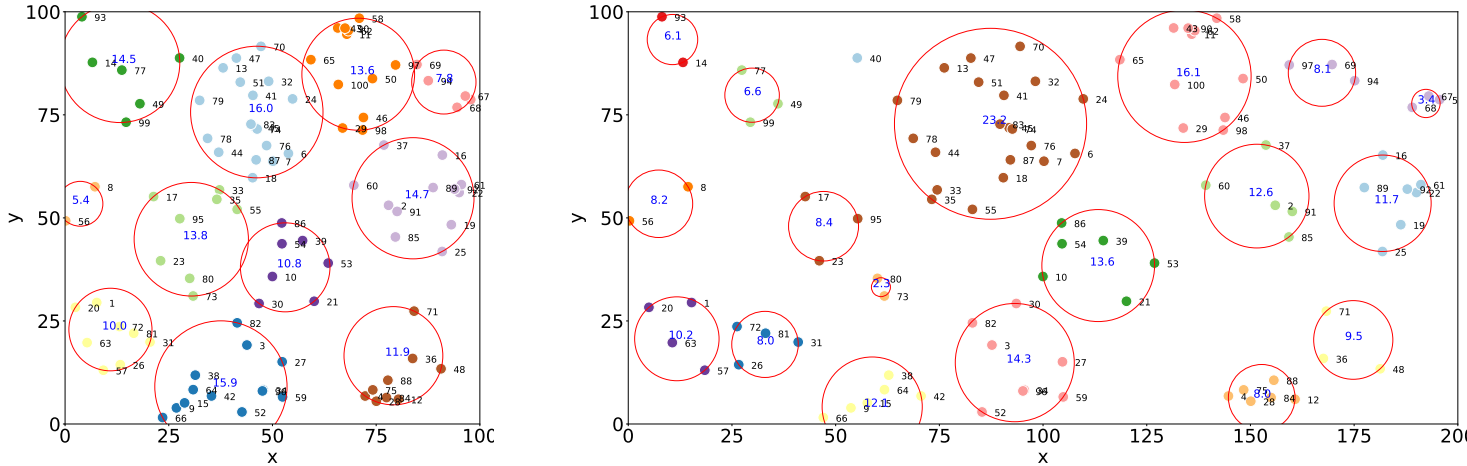
The problem of Berman et al. [14] is that the research compares the performances of the heuristics, so does not have decent benchmark solutions. For example, Berman et al. [14] reported that the GA could find the best-known solutions even for the large-sized problem classes, including 5,000 and 10,000 demand points. However, the proposed solutions have one UAV being operated, and every demand point is covered by the UAV, which is unrealistic. Even in the smallest problems, we found that the GA proposed by Berman et al. [14] could not find optimal solutions. Figures 3.3 and 3.4 (below) show the example solutions of the UVCP.



(a) B&P sol. of $1 \times 1, |N| = 20$

(b) B&P sol. of $2 \times 1, |N| = 50$

Figure 3.3: Example of the solution for small-sized problem.



(a) FRH-B&P sol. of $1 \times 1, |N| = 100$.

(b) FRH-B&P sol. of $2 \times 1, |N| = 100$.

Figure 3.4: Example of the solution for large-sized problem.

3.6.2 Solution algorithms

In the computational experiments, the GA proposed in Berman et al. [14] is used as a benchmark. To minimize the information held on each chromosome, the chromosome only consists of the position of the UAV and does not contain information of the allocation decision. The GA consists of three parts–local optimization, neighborhood search, and mating and evaluation. The local optimization part deletes the UAV that might not be used and relocates the UAVs to the local-optimal positions. Based on the current positions of the UAVs, each demand point is allocated to the closest UAV, and the UAVs without any demand point are removed. The remaining UAVs are relocated by solving the continuous single facility location problem. The procedure iterates until there are no more UAVs changing their locations. The local optimization is incorporated when a new individual is generated, including within the initial population, within the neighborhood search, and within the mating and producing of offspring. The neighborhood search part also finds a possible removal of a UAV that improves the objective function. By randomly removing one UAV, the improvement of the objective function is tested. The mating for a new offspring is done by merging two individuals and the repetition of the neighborhood search to the local optimum.

The B&P algorithm, the FRH, and the hybrid algorithm using FRH in the B&P algorithm (FRH-B&P) are compared in the experiment. The B&P and the FRH-B&P are the exact algorithms, and the FRH is an approximation algorithm.

We used the hybrid approach for the B&P algorithm itself to speed up the CG procedure. If the maximum coverage distance was limited, the feasible region of the subproblem II decreased, and so did the computation time. Also, when the CG was

solved to the optimal, the solution tended to have only a limited number of UAVs covering too large an area. In the early stage, the coverage distance was limited to the realistic size in the subproblem II. We denoted this as a limited-coverage subproblem. The limited-coverage subproblem contained an additional constraint, $r \leq R_{lim}$. When the limited-coverage subproblem could not find a solution with a negative reduced cost, the subproblem II without the limit of the coverage distance was used in the CG. For convenience, we denote the original subproblem II as a full-coverage subproblem in this section.

Because the limited-coverage subproblem provided the lower bound in the CG, if the lower bound and the primal bound satisfied the termination condition, the CG of the node could be terminated without solving the full-coverage subproblem. In the experiment, there were many instances in which the CG with the limited-coverage subproblem generated most of the columns required. In those cases, after the switch to the full-coverage subproblem, the full-coverage problem could not find the improving columns, or the CG was finished after a few iterations. This tendency and the proper value of the R_{lim} might be changed when a different objective function is employed. The CG algorithm of the hybrid approach is presented in Figure 3.5 (below). Note that the CG of the FRH-B&P shares a similar structure if the FRH replaces the limited-coverage subproblem.

In the problems of $|N| = 50$ and 100, we generated the initial columns based on the solution of the GA. An individual of the GA could be translated to several columns by allocating each demand point to the nearest UAV. After 200 generations or 200 seconds of the GA, 20 best individuals were generated as the initial columns and were included in the FRH and the FRH-B&P for the fast convergence.

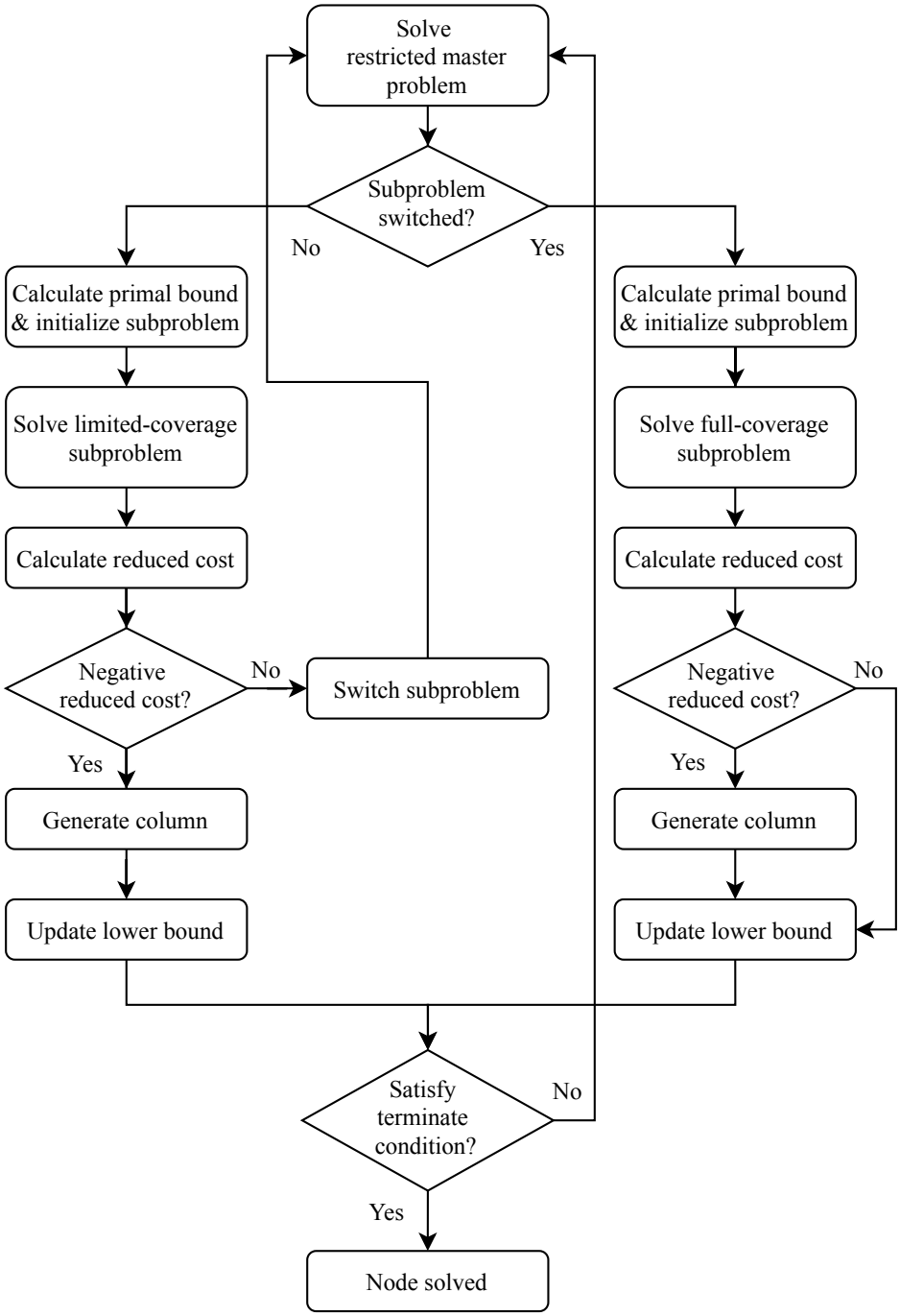


Figure 3.5: The procedure of the column generation algorithm with a hybrid approach

Furthermore, the restricted master heuristic (RMH) was used for the large-sized problems that could not find the optimal solution within the time limitation. The RMH is one of the most widely used heuristics related to the B&P algorithm. Based on the current variables (columns), one could solve the problem with the extended formulation using the MILP solvers. This algorithm is also called the *price-and-branch*, especially when it is executed after the termination of the root node CG, because the columns are generated first, and the branching takes place later. The UVCP does not consider the capacity constraint and assumes homogeneous UAVs. Thus, the RMH solved a small-sized, basic set partition problem and did not require a computation time longer than 0.01 seconds, even for the largest problems we experimented with. For the comparison of the RMH with other primal heuristics, one can refer to Section 4.4.6.

3.6.3 Algorithmic performances

We compared the performances of the B&P, the FRH, and the FRH-B&P. The GA proposed by Berman et al. [14] is used for the benchmark. The limitation of maximum computation time was set to 3,600 seconds.

First, the algorithmic performances were compared based on the computation speed and the optimality. Second, a further analysis of the performances of the CG and the B&P was conducted. Third, the performance of the additional techniques, including the initial columns and the limited-coverage subproblem, were analyzed.

For the first analysis, the computation speed of the B&P algorithm, the FRH, and the hybrid FRH-B&P algorithm were compared against each other and also compared against the benchmark GA. For the instances solved within the time

limitation, the computation time was recorded, and for the unfinished instances, the final gap between the primal bound and the best lower bound was compared, in order to measure the convergence of the algorithm. The optimality of results could be resolved in two ways. First, for the problems solved to the optimal, the limitation of the approximation algorithm, FRH, could be shown. For the unfinished problems, the feasible solutions and the primal bounds found by each algorithm could be compared. For each algorithm, the number of instances that were solved to the optimal were counted. The algorithms that could be found to be the best feasible solution for the unsolved instances were identified, and the optimality gap was compared. Also, the number of UAVs operated in the best feasible solution was compared.

In Tables 3.1 – 3.3 (below), the results are summarized by the average of 10 instances of each problem class. The columns in these tables are defined as follows:

- Dim: the dimension of the xy-plane. 1×1 and 2×1 represent 100×100 and 200×100 planes, respectively.
- $|N|$: the number of demand points.
- $\#solved$: the number of the solved instances, not necessarily to the optimal.
- $Time$: the average of the computation time. For the problems not solved within the time limitation, the time limit was used to calculate the average, and was marked with an asterisk (*).
- Gap_L: the average of the gap between the primal bound (the best feasible solution: BFS) and the best lower bound (BB). Gap_L was used to evaluate the algorithm's convergence speed for the instances not solved within the time

limit.

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

- $\#Opt$: the number of instances for which the algorithm provided the optimal solution.
- $\#Best$: the number of instances for which the algorithm provided the BFS, including the optimal solution.
- Gap: the average of the gap between the objective value of the BFS among every algorithm and the current algorithm.

$$Gap = \frac{(\text{obj. value of an algorithm}) - (\text{obj. value of the BFS})}{(\text{obj. value of the BFS})} \times 100\%$$

- $\#UAV$: the average of the number of UAVs operated in the BFS.

As mentioned in Chapter 2, the rapidly changing environment of UAV operation demands fast decisions. Table 3.1 shows that the proposed algorithms outperform the benchmark GA from the standpoint of computation speed. For the smallest problems of $|N|$ =10 and 20, the B&P algorithm without any heuristic and GA initial columns could solve the problem with the fastest computation time among the algorithms while providing the optimal solution. The B&P solved 33 out of 40 of the smallest problems within one second, which enabled the algorithm to be used for rapidly changing situations. Every solution algorithms could solve most of the small-sized instances with less than 50 demand points. Even though the possible number of the minimal subset had the dimension of $|N|^3$, and Berman et al. [14] anticipated that the exponential number of the power set had to be checked, in

the optimal solution, only a limited number of the columns were used. Because of the strong LP bound of the extended formulation of the B&P algorithm, the CG required only a minimum number of iterations to find the optimal solution. One of the essential characteristics of the UVCP is that the CG provided the integer solutions. That means that the branching is not required while solving the problem, a point that will be introduced in the second analysis, with Table 3.4 (below).

Because the FRH and the FRH-B&P algorithms are proposed for large-sized problems, we focused on the problem class of $|N| = 100$ when comparing the performances of those algorithms. Even though the exact algorithms, the B&P and the FRH-B&P, could not solve the large-sized problems within the time limits, the FRH could solve the eight instances within the time limits. In Table 3.2, it can be observed that the FRH could find the best feasible solution from ten out of 20 instances and that the FRH-B&P algorithm could find the BFS from 18 instances. Furthermore, the *gap* between the BFS and the FRH was less than one percent, which shows the effectiveness of the initial columns and the FRH as heuristics to provide a feasible solution. Still, to ensure or measure the optimality, the following B&P algorithm was required. Moreover, there are instances in which the FRH-B&P converged almost to the optimal. For example, in four instances of the 2×1 dimension, the Gap_L were lower than 0.001 percent. The *Gap* between the optimal solution and the solution of the FRH decreased as the size of the problem increased. This meant that the limitation of the approximation algorithm could be diluted when solving a large-sized problem.

Even though the GA could find the optimal solutions for some small-sized problems, it required a longer computation time for the convergence. The GA might

have found the optimal solution earlier than the finished time reported in Table 3.1. However, because the GA did not have a lower bound to measure the true convergence of the solution algorithm, it required additional computation to distinguish the termination. Thus, it was observed that in the small-sized problems, the B&P algorithm without any initial columns and heuristics could be the best choice that could solve the optimal solution in a short computation time. For the large-sized problems of $|N| = 100$, the FRH-B&P algorithm with the GA initial columns could be the best choice. If the computation time provided a hard limitation while solving the large-sized problem, the FRH with the GA initial columns could provide time-efficient solutions.

In Chapter 2, it was observed that the B&P can solve more *sparse* USCP in the shorter computation time. The problems with the 2×1 dimension have a more sparse distribution of the demand points than the 1×1 dimension problems. The similar structure of the UVCP allows the faster convergence of the B&P in the 2×1 dimension, which is observed in Table 3.1. Table 3.3 shows the number of UAVs operated in the solution of each algorithm. There was no distinct tendency in how the number of UAVs was related to the optimality gap. However, when solving the large-sized problem with the B&P algorithm, if the CG was not converged, then there were cases in which the generated columns had small coverage distances, which caused a large number of UAVs to be operated compared to other algorithms. We also could see the result that under this cost structure, the number of UAVs provided a larger impact than a larger coverage area. A more detailed analysis of the number of UAVs in relation to the operating objective, along with more realistic cost information, could be helpful when implementing the UAV system in real applications.

Table 3.1: Results related to the computation speed

Dim	$\|N\|$	*#solved*				*Time(s)*				$Gap_L(\%)$		
		B&P	FRH	FRH-B&P	GA	B&P	FRH	FRH-B&P	GA	B&P	FRH	FRH-B&P
1×1	10	10	10	10	10	0.12	0.55	0.68	70.04	0.0	0.0	0.0
	20	10	10	10	10	1.14	7.02	8.38	517.44	0.0	0.0	0.0
	50	10	10	9	9	754.69	308.31	1,105.51*	2,679.13*	0.0	0.0	< 0.05
	100	0	8	0	0	3,600*	2,337.18*	3,600*	3,600*	213.5	0.1	114.1
2×1	10	10	10	10	10	0.08	0.43	0.52	56.21	0.0	0.0	0.0
	20	10	10	10	10	0.63	2.16	2.85	367.43	0.0	0.0	0.0
	50	10	10	10	8	84.43	162.73	229.85	1,933.28	0.0	0.0	0.0
	100	0	8	0	0	3,600*	1,623.80	3,600*	3,600*	79.5	0.0	7.4

*: There are instances not solved within the time limit.

Table 3.2: Results related to the optimality (objective value)

Dim	$\|N\|$	*#Opt/#Best*				*Objective value*				*Gap*(%)			
		B&P	FRH	FRH-B&P	GA	B&P	FRH	FRH-B&P	GA	B&P	FRH	FRH-B&P	GA
1×1	10	10/10	5/5	10/10	10/10	823.2	862.7	823.2	823.2	0.0	5.4	0.0	0.0
	20	10/10	3/3	10/10	9/9	1,375.1	1,486.4	1,375.1	1,375.8	0.0	8.4	0.0	0.1
	50	10/10	4/4	10/10	2/2	2,312.1	2,321.8	2,312.1	2,332.4	0.0	0.4	0.0	0.9
	100	0/0	0/7	0/9	0/2	3,645.2	3,059.9	3,035.4	3,092.0	19.3	0.1	< 0.05	7.4
2×1	10	10/10	9/9	10/10	10/10	911.2	916.0	911.2	911.2	0.0	0.6	0.0	0.0
	20	10/10	5/5	10/10	10/10	1,668.7	1,696.2	1,668.7	1,668.7	0.0	1.7	0.0	0.0
	50	10/10	10/10	10/10	7/7	3,084.8	3,084.8	3,084.8	3,095.5	0.0	0.0	0.0	0.4
	100	0/0	0/3	0/9	0/2	4,590.5	4,419.8	4,064.4	4,436.1	4.5	0.7	< 0.05	1.1

Table 3.3: Results related to the optimality (# of UAVs)

Dim	$\|N\|$	#UAV			
		B&P	FRH	FRH-B&P	GA
1×1	10	7.0	7.0	7.0	7.0
	20	9.5	8.6	9.5	9.5
	50	11.8	12.2	11.8	12.9
	100	15.4	12.7	12.8	12.5
2×1	10	8.3	8.2	8.3	8.3
	20	13.5	13.3	13.5	13.7
	50	20.4	20.4	20.4	20.2
	100	23.8	22.6	22.9	23.7

For the second analysis, the number of iterations and the number of generated columns were compared through the B&P and the heuristic-related algorithms. The branching and the nodes in the branching tree were also analyzed. In the B&P algorithms, the basic initial columns were generated, where one UAV covered one demand point. Thus, the cost of the basic initial column only consisted of the fixed cost of a UAV, 100. The FRH-B&P algorithm started with the FRP. After the FRP was finished, the full B&P algorithm was started, based on the columns generated in the FRP. The number of iterations and the columns of the FRP-B&P were the summation of the one executed in the FRP and the following B&P. The columns in Table 3.4 are defined as follows:

- $\#Nodes$: the average number of the nodes generated in the branching trees. For the problems not solved within the time limitation, the current value was used to calculate the average, and was marked with an asterisk (*).
- $\#Iter$: the average number of the iterations executed in the CG and the B&P

algorithm.

- $\#Columns$: the average number of the columns generated in the CG and the B&P algorithm.

In the experiment, it was observed that for the exact algorithms of the B&P and the FRH-B&P, the root node CG provided the optimal integer solution. Thus, without any corresponding branching, the optimization algorithm was terminated. In Table 3.4, only the FRH had the column of the average number of nodes generated in the branching trees. Furthermore, even the FRH had the small value of $\#Nodes$. Compared to the results in the USCP, this identified one of the most important characteristics of the UVCP.

The difference of the nodes between the UVCP and the USCP did not originate from the structure of the objective function, which shows that the branching still happened in the FRH. Also, unlike in the EBP of the USCP, the columns provided by the FRH were still "tight." Therefore, it is difficult to say that the columns that are not tight caused the difference.

The only possibility left is that the subproblem of the UVCP solved the CG to the optimal without any restriction of the coverage distance. As mentioned in Berman et al. [14], there exists an optimal solution of the UVCP that is a union of the minimal subsets. Because each minimal subset corresponds to a column in the extended formulation, there exists an optimal solution of the UVCP that is a union of the columns. Our conjecture is that in the UVCP, both the fractional solution and the negative reduced cost are related, and if the restricted master problem has every column used in the optimal solution of the UVCP, then the subproblem does not have a new column with a negative reduced cost. Also, every minimal subset is

feasible in the subproblem, so the root node CG will be able to find every minimal subset required in the optimal solution of the UVCP. This characteristic of the UVCP highlights the advantages of the speed-up techniques of the CG.

As mentioned above, the FRH-B&P solved the original B&P algorithm after finishing the FRH. Therefore, in the FRH-B&P, the iterations already executed in the FRH were added when calculating its number of iterations. In the problem class of $|N| = 50$, the number of generated columns in the FRH-B&P was larger than the number generated in the B&P algorithm, even though the FRP-B&P required fewer numbers of iterations. This was originated by the GA initial columns, which were generated before the FRH. The FRH, admittedly, can provide efficient columns rapidly; those columns are not the columns with the most negative reduced cost, which are solved in the original pricing subproblem. Thus, the number of columns generated in the FRH and the FRH-B&P tended to be larger than in the B&P.

As mentioned above, it was easier to solve the sparse problems, so the problems with the 2×1 dimension required fewer iterations than the problems with the 1×1 dimension, and required fewer columns. Note that the B&P algorithm required only a minimum number of columns while solving the problems to the optimal. Compared to the total number of the minimal subset, which had the dimension of $|N|^3$, the number of the iterations and the columns stayed less than 1,000, even for the largest problems. Thus, when the RMH was used for the largest problems that were not solved within the time limit, the computation time of the RMH stayed less than 0.01 seconds for every problem.

For the third analysis, additional techniques to speed up the computation were analyzed. As mentioned in Section 3.6.2, the FRH and the FRH-B&P started with

Table 3.4: Results related to the CG & branching

Dim	$\|N\|$	*#Nodes*	*#Iter*			*#Columns*		
		FRH	B&P	FRH	FRH-B&P	B&P	FRH	FRH-B&P
1×1	10	1.0	5.2	4.5	9.9	13.3	13.9	17.3
	20	1.0	15.5	17.2	32.1	33.5	37.1	50.0
	50	1.0	131.2	77.1	125.1*	179.2	154.9	201.4*
	100	1.1*	192.2*	572.7*	631.8*	292.2*	733.6*	765.8*
2×1	10	1.0	3.7	2.9	6.8	11.7	12.0	13.9
	20	1.0	9.1	8.9	18.0	27.1	28.4	35.5
	50	1.0	50.4	27.8	50.7	98.4	104.0	124.9
	100	2.3	160.0*	204.8	256.6*	260.0*	380.3	432.1*

*: There are instances not solved within the time limit.

the initial columns generated by the GA for the problems of $|N| = 50$ and 100. Although the GA required a long computation time to find good individuals, the embedded heuristic inside of the GA could help find the local optimal subset of demand points. A solution of the GA was decoded into the mathematical model and divided into multiple columns to be included in the solution algorithms. In the analysis, the B&P with the GA initial columns was also tested to see the improvement of the lower bound.

Also, the effect of the limitation of the coverage was analyzed. In the hybrid algorithm of the limited and full-coverage subproblems, the time spent for each subproblem, along with the number of iterations, was compared. The columns in Tables 3.5 and 3.6 (below) are defined as follows:

- B&P w. clm.: the B&P algorithm starting with initial columns generated by the GA.
- $Time_{lim}$: the average of the computation time spent on the limited-coverage

subproblem. For the problem class in which the limited-coverage subproblem was not finished within the time limitation, the time limit was used to calculate the average, and was marked with an asterisk (*).

- $\#Iter_{lim}$: the average of the number of iterations executed in the limited-coverage subproblem.

By generating multiple initial columns with the GA, the B&P could improve both primal and lower bounds when solving the large-sized problems. The effective initial column decreased the number of iterations required for the column generation. However, for the small-sized problems, the advantages did not show up because the B&P algorithm without initial columns did not require a large number of iterations by itself. In Table 3.6, the difference between the number of iterations was not significant in the problem class of $|N| = 10$ and 20. The effective GA initial columns contained the columns covering a larger distance than the limit set in the limited-coverage subproblem. In the B&P algorithm using a hybrid approach without initial columns, when the size of the problem exceeded a certain level, it became harder to solve the last few iterations of the full-coverage subproblem. This originated from two reasons. First, the full-coverage subproblem took a longer computation time than the limited-coverage subproblem. Second, it became more challenging to solve the subproblem when the CG algorithm converged. Thus, in the problem class of $|N| = 50$, the ratio between $time_{lim}$ and $time$ stayed very low. Especially in the instances with the dimension of 2×1, it took an extra 30 seconds to execute one iteration of the full-coverage subproblem. However, the B&P with GA initial columns started with columns of a large coverage distance. Thus, the ratio of the computation time of the two subproblems stayed very high, meaning that there was less effort

required for handling the full-coverage subproblems.

In the problem class of $|N| = 100$, both B&P and B&P with GA initial columns could not solve the problem to the optimal. Therefore, the performances were compared based on the lower bound and the Gap_L. The advantages of the FRH have already been mentioned in the previous analysis. However, some of the advantages originated from the usage of the GA initial columns. The improvement of the lower bounds by each technique was observed, and the FRH-B&P, which used both techniques, performed the best.

The advantage of the hybrid approach of the limited-coverage and full-coverage subproblems was shown in the comparison of the number of iterations. In most of the iterations in the CG, only columns with the limited coverage radius were produced. Because the full-coverage subproblem took a longer computation time while providing the same solution, it was more useful to focus on the smaller feasible region.

Table 3.5: Results related to the lower bounds

Dim	$\|N\|$	*Lower bounds*			$Gap_L(\%)$
		B&P	B&P w. clm.	FRH-B&P	B&P w. clm.
1×1	10	823.2	823.2	823.2	0.0
	20	1,375.1	1,375.1	1,375.1	0.0
	50	2,312.1	2,304.0	2,312.1	0.4
	100	-4,142.9	-623.4	-407.2	120.5
2×1	10	911.2	911.2	911.2	0.0
	20	1,668.7	1,668.7	1,668.7	0.0
	50	3,084.8	3,084.8	3,084.8	0.0
	100	912.8	3,760.5	4,064.4	14.3

Table 3.6: Comparison of the hybrid algorithm and initial column

Dim	$\|N\|$	B&P			B&P w. clm.			B&P		B&P w. clm.	
		$Time_{lim}$	$Time$	$Ratio$	$Time_{lim}$	$Time$	$Ratio$	$\#Iter_{lim}$	$\#Iter$	$\#Iter_{lim}$	$\#Iter$
1×1	10	0.09	0.12	75.1	31.98	32.02	99.9	4.2	5.2	1.9	3.0
	20	0.93	1.14	81.1	127.11	127.35	99.8	14.5	15.5	6.7	7.8
	50	160.54	754.69	34.7	275.72	924.41*	54.3*	122.5	131.2	68.1	73.1*
	100	3,600*	3,600*	100*	3,600*	3,600*	100*	192.2*	192.2*	183.2*	183.2*
2×1	10	0.04	0.08	51.1	20.98	21.01	99.8	2.7	3.7	1.2	2.2
	20	0.41	0.63	63.5	63.42	63.65	99.5	8.1	9.1	2.7	3.7
	50	54.79	84.43	66.1	139.55	170.58	82.3	49.2	50.4	24.4	25.5
	100	3,600*	3,600*	100*	3,600*	3,600*	100*	160.0*	160.0*	123.8*	123.8*

*: There are instances that the limited-coverage subproblem is not finished within the time limit.

3.7 Summary

Based on the results of the USCP, the problem of developing a flight plan of a UAV while taking into account the variable coverage area was proposed in this chapter. The problem was modeled from existing problem that focused on the planar version of the variable radius covering problem, which is defined by Berman et al. [14].

The UVCP was modeled as a set covering problem without a limitation of the predefined candidate position and with an extra decision of the variable coverage distance. The mathematical model was defined in an explicit formulation. Due to the nonconvex feasible set of the continuous relaxation, the problem was not solvable by the optimization software. An extended formulation and a corresponding B&P algorithm were proposed to utilize the concept of the minimal subset. Based on the minimum covering circle, we reformulated the subproblem into the solvable mixed-integer linear quadratic programming model and proved the equivalence of two formulations. Based on the equivalence, it was also shown that the UVCP can be solved by Dantzig-Wolfe decomposition and the B&P algorithm. The fixed-radius heuristic solved multiple fixed-radius problems of the UVCP, which had the same feasible set of the USCP. A hybrid approach using FRH and the original B&P was developed to accelerate the computation speed of the large-sized instances. The computational results showed that the proposed B&P algorithms outperformed the benchmark GA proposed by Berman et al. [14] whenever they could find the optimal solution with the shorter computation time for the small-sized problems. The proposed B&P algorithms had better primal bounds for the large-sized problems and could measure the convergence of the optimization by the gap between the primal and the lower bounds. One characteristic observed in the computational experiment

was that the root node column generation of the B&P algorithm provided an integer solution, so that the branching was not required in the UVCP. That characteristic emphasized the need for the fast convergence of the CG, which could be accelerated by the effective initial columns, heuristics, and hybrid approaches proposed in this chapter.

Chapter 4

Facility location-allocation problem for unmanned aerial vehicle emergency medical service

4.1 Introduction

The medical system plays an essential role in modern society by protecting human life and health. Among the various components that constitute the medical system, the emergency medical service (EMS) system is important in connecting communities directly to the healthcare system. As a point of contact between patients needing urgent care and the medical system, the EMS provides basic diagnosis and treatment services to patients in the field or on the road while transporting them to the hospital. The EMS consists of transport, trained professionals, and medical equipment. The development of and investment in the EMS system have focused on increasing the number of resources and implementing advanced equipment. Expanding from the transport-only EMS, which only transported patients to the hospital, the EMS system has augmented its services with first aid equipment, basic medications, defibrillators, and emergency oxygen tanks in basic life support (BLS) ambulances. It also has added intravenous treatments, electrocardiograms, airway intubations, and capnography in advanced life support (ALS) ambulances. In recent years, air medical services operated by physicians, nurses, and paramedics with ALS ambulance have

been used in remote areas with sparse populations or in complex metropolises with heavy traffic. However, there is a limit to expanding the number of vehicles and the human resources invested in EMS. Thus, modern society's healthcare system tries to spread primary medical devices widely in public areas. One example of this is the automated external defibrillator (AED) placed in public areas. AEDs regulate the heartbeats of individuals experiencing sudden cardiac arrest and play a central role in the "chains of survival." Because rapid intervention with AEDs can secure the survival rate of patients [70], AEDs can be used to extend the "golden hour" of ambulance arrival. Thus, AEDs have been widely installed in modern cities. For example, in Korea, there is a law to keep AEDs in every building and transportation facility with a large number of people.

In the company with the development of UAV technology, several attempts have been made to utilize UAVs in public security and healthcare services [120, 94, 109]. In the same manner of the commercial UAV-operated logistics [68, 74], UAVs can be operated as multipurpose emergency medical resources providing rapid and flexible responses. It is difficult to maintain a responsive EMS system in remote areas, such as hamlets and isolated dwellings, and in city centers with heavy traffic [29]. However, if UAVs can augment the EMS system, they can be used to access demand points through the air, thereby avoiding traffics. Multiple projects are trying to implement UAVs to search for and identify patients, provide information [94], and transport emergency medicine, blood, and AEDs. One of the most rapidly developing areas is a UAV EMS (UEMS) known as an "ambulance drone." A UAV with a built-in AED can reach sudden cardiac arrest patients faster than an ambulance. As introduced above, making defibrillators publicly available boosts survival rates. For detailed

information on the necessity of UEMS and the current status of the implementation, one can refer to Wankmüller et al. [109].

To learn how the EMS system can operate more efficiently, researchers have been related the EMS location to the facility location models. Daskin [33] subdivided facility location models into four categories–*analytic*, *continuous*, *network*, and *discrete models*–based on the problem's space. As emphasized in Daskin [33], the discrete models are dominant for healthcare facility locations because they focus on the discrete decisions of opening, operation, and assignments of demand points for the facilities. There are two approaches to the discrete facility location models for the healthcare facility planning, which are introduced in Section 1.2. In the covering-based approach, under the given candidate locations of facilities and demand points, the feasibility of a coverage between a facility and a demand point can be considered as binary. In the covering problem, the actual distance is binarized, and the demand point is considered to be "covered" if it is in the critical coverage distance from a facility. Thus, the distance between the facility and the demand point is considered as a constraint of the problem.

In the median-based approach, the distance between the facility and the demand point is considered as a cost in the objective function, instead of a constraint. In other words, there is no limitation on the coverage distance of a facility. Any demand point can be assigned to a facility as a decision. One of the famous models of this concept is the location-allocation problem, which decides the facility's opening and the assignment of each demand point to that facility. The distance-weighted cost is accumulated to the objective function, in the company with the facility's opening cost. The main difference between the two approaches is based on the way the

distance between a facility and a demand point is considered. In the covering-based approach, the distance is binarized with the coverage distance limitation of a facility. Conversely, in the median-based approach, the demand point is allocated without any restriction, as long as the cost is valued as high enough.

However, when operating multiple UAVs in the UEMS system, the existing approaches cannot apply the characteristics of UAVs. One of the major characteristics is the physical limitation of a UAV's flight distance. Due to its payload limitation, UAVs can only cover a bounded area around the facility [109, 68, 74, 67, 57]. Another characteristic is that multiple UAVs are operated in one facility within the UEMS, while only one or two ambulances are operated in one facility in the ordinary EMS system. When considering the bounded coverage distance in the existing literature, the problem is modeled with a covering-based approach. In that case, resource capacity and availability are hardly considered, so it is hard to model the effect of the number of UAVs assigned to one facility. If a location-allocation model is employed, the resource availability affected by the distribution of the demand points is considered in the cost parameter, instead of a constraint.

To locate the UEMS facility and assign multiple UAVs efficiently to facilities, the coverage distance limitation has to be considered in detail, so that the number of UAVs determines the resource availability. From the classical approaches of maximum expected covering location model [32] and maximum availability location problem [88], the *busy-fraction* of one unit of the resource was used to calculate the total resource availability by measuring the chance that at least one resource is available. With the predefined coverage area and the busy-fraction of each resource, one could calculate the expected amount of the covered demand and, on the con-

trary, the number of the resource required to fulfill the α-*reliability*. We approached the busy-fraction concept from another angle. When the number of the resource increases, the coverage area can be also increased without the loss of the reliability, regardless of the increased busy-fraction. In the literature of operating UAV system, Shakhatreh et al. [96] showed that if the UAV requires a setup such as recharge between flights, the coverage distance of the system increases along with the number of UAVs.

The decision of the coverage area is modeled as the variable coverage distance in this research, and the number of UAVs assigned to a UEMS facility define the coverage distance as well as the capacity of the facility, which is introduced for the first time in this research. The location and operation problem of the UEMS system is defined and named as a UEMS location-allocation problem (ULAP). The ULAP includes the characteristics of both covering-based and median-based approaches. The variable coverage distance constraint is modeled as a quadratic function based on the proximity of resource availability and the size of the covered area, and then reformulated into the equivalent linear formulation. Also, the allocation decision of the uncertain demand is considered with the capacity of the UEMS facility, as in the median-based approaches. A cost-minimization problem, while fulfilling every demand of the UEMS system, is modeled with a robust optimization approach.

In the robust optimization approach, the demand is modeled with the cardinality-constrained uncertainty set, and the nonlinear capacity constraints are linearly reformulated. However, the reformulation model contains integer and continuous decision variables and has highly fractional solutions in the LP relaxation of the problem. Because of the weak LP relaxation bound, the commercial optimization solvers

have difficulty solving realistically-scaled problems. If the original problem is decomposed into smaller problems, taking into account individual facilities, each decomposed problem is related to the robust and integer knapsack problems. To utilize the knowledge of the knapsack problem, an extended formulation (EF) based on the Dantzig-Wolfe decomposition is proposed and solved by the branch-and-price (B&P) algorithm. The EF provides a better LP relaxation and decreases symmetries of the branching tree. With the B&P algorithm, the subproblem was solved by two approaches–mixed-integer linear programming (MILP) reformulation and decomposed dynamic programming (DP) approaches–each of which has its own advantages. Furthermore, a restricted master heuristic based on the B&P algorithm is proposed to provide a time-efficient feasible solution to large-sized problems.

The rest of this chapter is structured as follows: Section 4.2 briefly introduces literature related to the healthcare facility location problem. Section 4.3 proposes the problem definition, mathematical formulation, and linear reformulation of the mathematical model. Section 4.4 presents detailed information on the B&P algorithm of the ULAP. The extended formulation, branching strategies, structure of the subproblem, and the solution approaches of the subproblem are introduced. In addition, the restricted master heuristic for the primal solution is presented. The proposed algorithms are compared with the computational experiments in Section 4.5. Finally, Section 4.6 concludes the chapter.

4.2 Related literature

Previous literature related to the ULAP is introduced in this section. As mentioned above, there are covering-based and median-based approaches in the healthcare fa-

cility location problem. The coverage distance used in the covering-based approach was quantified by Daskin [32] based on the term *busy-fraction*, which measures the availability of the resource. ReVelle and Hogan [88] measured the site-specific busy fraction of a particular site. To overcome the limitation of the digitized all-or-nothing coverage distance, a concept of double coverage [47] was proposed, which forced the system to cover every demand point within a large radius and, at the same time, to cover a certain proportion of the demand point within a small radius. Backup coverage [55] maximizes the number of demand points covered by more than two facilities while covering every demand point at least once.

One limitation of the covering-based approach is the disregard of the quantity of the demand and the facility's capacity, which requires the allocation decision of the demand points to the facilities as in the location-allocation approach. In the recent works of the location-allocation approach, the demand uncertainties are investigated thoroughly with the stochastic and robust models considering the probabilities of the demand satisfaction [118]. The intractabilities of the stochastic and robust models were tackled by limiting the set of feasible facilities for each demand point or fixing policies for allocations [12]. We refer readers to the recent research of Bertsimas and Ng [16], who reviewed the probabilistic models of ambulance deployment and modeled ambulance deployment with recourse actions. In the location-allocation approach, the distance between the facility and the demand point is considered in the objective function or only filters the impossible pairs of the facility-demand point, as in Wankmüller et al. [109]. Although the existing location-allocation approaches consider probabilistic constraints of the demand satisfaction, resource availability is not yet related to the decision of the size of the covered area as constraints.

Two directions consider allocation decisions in the company with the covering-based approach. In the gradual cover model [15, 41, 60, 16], the demand of each demand point is "partially" satisfied by a function that decays according to the distance from a facility. In the gradual cover model, the partial satisfaction of demand [15] or the quality of service [41] are modeled in the objective function. Even though the allocation is considered as decisions in the gradual cover model, the coverage distance of a facility is not modeled as constraints and cannot be treated as decisions. Another direction is the variable cover model [14]. The coverage distance of a facility is treated as a decision variable, and the cost of a facility is decided by a monotonically increasing function of a coverage distance. However, the variable cover model is still bounded in the covering-based approach because the facility's capacity and the size of the demands are not considered. Thus, the demand uncertainties are hardly considered, and it is relatively simple to provide efficient feasible solutions with heuristics and metaheuristics.

To the best of the authors' knowledge, there is one existing study related to the capacity and the coverage distance [3]. However, Akl et al. [3] determined the capacity of a wireless network and used this to allocate clients to facilities, so the coverage distance and the capacity were assumed to be related inversely. Thus, the solution approaches proposed in [3] cannot be used in the ULAP. For literature focusing on healthcare facility location problems, we refer readers to the related review papers [13, 10, 1]. Table 4.1 compares this research to the existing literature.

Table 4.1: Comparison of this research and existing literature

Author (year)	**Type**	**Coverage distance constraint**	**Demand**
Gendreau et al. (1997) [47]	Set cover	double coverage	binary
Hogan and ReVelle (1986) [55]	Set cover	backup coverage	binary
ReVelle and Hogan (1989) [88]	Set cover	single coverage	binary
Choi and Chaudhry (1993) [28]	Median	single coverage	binary
Calik et al. (2015) [23]	Median	unconstrained	binary
Eiselt and Marianov (2009) [41]	Set cover, Median	gradual coverage	partial
Jang and Lee (2015) [60]	Median	gradual coverage	partial
Beraldi and Bruni (2009) [12]	Median	unconstrained	uncertain
Zhang and Jiang (2014) [118]	Median	unconstrained	uncertain
Zhang and Li (2015) [119]	Median	unconstrained	uncertain
Bertsimas and Ng (2019) [16]	Median	unconstrained	uncertain
Daskin and Maas (2015) [34]	Median	unconstrained	binary
Wankmüller et al. (2020) [109]	Median	single coverage	deterministic
Berman et al. (2009) [14]	Set cover, Median	variable coverage	deterministic
Akl et al. (2015) [3]	Median	variable coverage	deterministic
This dissertation (Chapter 4)	Set cover	variable coverage	uncertain

4.3 Location-allocation model for UEMS facility

This section provides a detailed definition of and assumptions about the UEMS location-allocation problem. To relate the known information of demand and its uncertainty to resource availability, the UEMS is modeled as a location-allocation problem with uncertain demands. Also, the allocation of a demand point to a facility is constrained with a variable coverage distance of the facility, which is bounded with the number of UAVs assigned to a facility. The number of UAVs assigned to a facility determines the capacity and the coverage of the facility with a positive relation. The proposed ULAP provides a strategic operation plan of a UEMS system, which consists of the decision of opening the UEMS facilities, the number of UAVs assigned to each UEMS facility, and the allocation of demand points to the facilities.

4.3.1 Problem definition

The ultimate goal of the UEMS system is to respond to medical emergencies while incurring minimum casualties. Thus, the ULAP tackles the demand uncertainty with a robust approach. The demand uncertainty is modeled on the well-known cardinality-constrained uncertainty set, which is proposed by Bertsimas and Sim [18]. In the cardinality-constrained uncertainty set, the information of the demand is given with a nominal value and symmetric distribution that is bounded as an interval. The approach focuses on the robust decision, which protects the feasibility of the solution against the targeted number of perturbations.

In the ULAP, the number of UAVs assigned to a facility defines the capability of the facility in two ways: (1) the total capacity and (2) the coverage distance, and restricts the allocation of the demands as in Akl et al. [3]. Like ordinary capacitated facility location problems, the capacity limits the accumulated demands. Simultaneously, the capacity of a facility has to consider the individual demand point because the capacity limits the maximum distance between the facility and the allocated demand points.

While modeling the resource availability based on the variable coverage distance, we followed the basic concept of the variable coverage distance proposed by Berman et al. [14] and Akl et al. [3]. The method of measuring the resource availability based on the busy-fraction and the number of the resource has already been used before in the maximum availability location problem [88]. In this research, the variable coverage distance is modeled as a maximum coverage distance assuring a certain level of resource availability. Given the assumption that the demand is evenly distributed over the plane, the amount of covered demand increases along with the expansion of

the covered area. To maintain the same resource availability, the required capacity of the facility must increase quadratically when increasing the coverage distance. These are comprised in the mathematical models in Section 4.3.2.

The assumptions of the presented problem are defined as follows:

(1) The locations of the demand point and candidate facility are already known.

(2) The facility's opening cost and the purchase and operation cost per UAV of the UEMS facility are known.

(3) The coverage distance of a facility is defined based on the number of the UAVs assigned to the facility.

(4) The capacity of a facility is defined based on the number of UAVs assigned to the facility.

(5) Every demand has to be satisfied.

(6) Each demand point has to be allocated to a certain facility. In other words, a partial allocation of a demand to multiple facilities is prohibited.

(7) There is an upper limit to the number of UAVs assigned to a facility.

(8) A demand is known with a nominal value and a value of the maximum perturbation. The uncertain demand obeys symmetric distribution, where the support is given as an interval.

Figure 4.1 presents an overview of the ULAP. The objective of the ULAP is to minimize the total cost to fulfill every demand while also considering uncertainties. The capacity of a UEMS facility is decided by the assignment of the UAVs, which

requires the opening of the facility. The demand uncertainty is protected in the capacity constraint with a limited conservatism. That is, the ULAP considers the capacity constraint while limiting the number of the demand points with maximum perturbations. The uncertainty of the *cardinality* of the demand points is considered for each UEMS facility. Thus, the authority can control the *conservatism* of the individual facility.

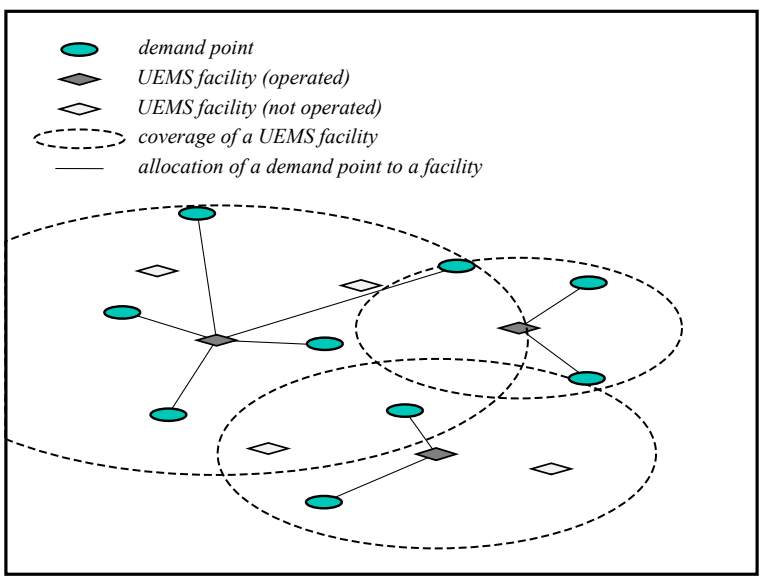


Figure 4.1: Overview of the ULAP.

4.3.2 Mathematical formulation

The following notations are used to formulate a mathematical model of the ULAP.

Sets

I set of candidate UEMS facilities.

J set of demand points.

Parameters

f_i fixed cost of opening UEMS facility i.

p_i assignment cost per unit UAV at UEMS facility i.

n_i maximum number of UAVs operated at UEMS facility i.

r_{0i} minimum insured coverage distance of UEMS facility i.

μ_i variable coverage distance ratio per UAV of UEMS facility i.

s_{ij} distance between UEMS facility i and demand point j.

$\tilde{d}_j$ uncertain demand of demand point j.

Decision variables

$$y_i = \begin{cases} 1, & \text{if UEMS facility } i \text{ is opened.} \\ 0, & \text{otherwise.} \end{cases} \qquad \forall i \in I$$

$$x_{ij} = \begin{cases} 1, & \text{if demand point } j \text{ is allocated to UEMS facility } i. \\ 0, & \text{otherwise.} \end{cases} \qquad \forall i \in I, \ \forall j \in J$$

$$u_i \in \mathcal{Z}_+, \quad \text{number of UAVs assigned at UEMS facility } i. \qquad \forall i \in I$$

The information of the candidate UEMS facilities is given as parameters. It includes the amortized cost related to the facility and the UAV. Also, the upper limit of the UAVs to be assigned to each facility is known. In our model, the assignment of UAVs provides a minimum insured coverage distance around a facility, and the implementation of multiple UAVs increases the coverage distance with a given variable coverage distance ratio. The ULAP models the demand uncertainty based on the cardinality constrained uncertainty set. The demand, $\tilde{d}_j$, of a demand point, j, is modeled to take value in the interval $[d_j^0 - \hat{d}_j, d_j^0 + \hat{d}_j]$ with symmetric distributions, where d_j^0 is a nominal value and $\hat{d}_j$ is a maximum perturbation. $\Gamma_i \in [0, n_i]$ is defined

for each facility, i, that controls the level of conservatism. There are three types of decision variables: two binary decision variables of the ULAP and one integer variable deciding the capacity of a facility. The standard formulation (SF) is a *compact* formulation with a nonlinear constraint.

[Standard formulation]

$$\min \quad \sum_{i \in I} f_i y_i + \sum_{i \in I} p_i u_i \tag{4.1}$$

$$\text{s.t.} \quad n_i y_i \geq u_i, \qquad \forall i \in I \tag{4.2}$$

$$y_i \geq x_{ij}, \qquad \forall i \in I, \forall j \in J \tag{4.3}$$

$$u_i \geq \sum_{j \in J} \tilde{d}_j x_{ij}, \qquad \forall i \in I \tag{4.4}$$

$$r_{0i} + \sqrt{\mu_i u_i} \geq s_{ij} x_{ij}, \qquad \forall i \in I, \forall j \in J \tag{4.5}$$

$$\sum_{i \in I} x_{ij} \geq 1, \qquad \forall j \in J \tag{4.6}$$

$$x_{ij} \in \mathbb{B}, \qquad \forall i \in I, \forall j \in J \tag{4.7}$$

$$y_i \in \mathbb{B}, \qquad \forall i \in I \tag{4.8}$$

$$u_i \in \mathcal{Z}_+, \qquad \forall i \in I. \tag{4.9}$$

Constraint (4.2) relates the assignment decision of UAVs to the opening of the UEMS facility and Constraint (4.3) links the location (opening) and allocation decisions. Constraint (4.6) ensure every demand to be covered. Constraint (4.4) bounds the allocation of the demand to the capacity of the facility. The value of the demand is calculated as the number of UAVs required. The uncertainty of the demand is not described in the controllable format yet. In the cardinality constrained uncertainty

set, we optimize the objective function against every scenario in which the uncertain demand is realized, as long as the number of demand points with extreme perturbation is lower than the target protection level Γ_i. According to Bertsimas and Sim [18], Constraint (4.4) can be modeled for each facility, i, as:

$$\begin{aligned} u_i \geq & \sum_{j \in J} \tilde{d}_j x_{ij} \\ = & \sum_{j \in J} d_j^0 x_{ij} + \max_{N \subseteq J, |N| = \Gamma_i} \sum_{j \in N} \hat{d}_j x_{ij} \end{aligned} \tag{4.4'}$$

Note that Constraint (4.4′) is still nonlinear. Constraint (4.5) is a quadratic constraint that defines the variable coverage distance around a facility. The coverage distance is defined as a summation of a minimum insured coverage distance and the variable distance related to the number of UAVs and the variable coverage ratio. Although we modeled the variable coverage as a quadratic constraint because of its simplicity, other monotonic non-decreasing (e.g., linear or piecewise-linear) functions can be utilized instead of the quadratic function. One can refer to the literature related to resource availability and the busy-fraction [71, 87, 102, 60].

4.3.3 Linearization of the quadratic variable coverage distance function

Let a parameter CV_{ij} denote the coefficient of coverage distance between facility i and demand point j:

$$CV_{ij} := \begin{cases} 0, & \text{if } r_0 \geq s_{ij}. \\ \dfrac{(s_{ij} - r_{0i})^2}{\mu_i}, & \text{otherwise.} \end{cases}$$

Considering Constraints (4.7) and (4.9), an equivalent linear reformulation of Constraint (4.5) is proposed:

Proposition 1. *The following inequality $u_i \geq CV_{ij}x_{ij}$ is equivalent to Constraint (4.5) for the ULAP.*

Proof. We only have to consider facility i and demand point j which satisfies $s_{ij} \geq r_{0i}$.

($\Rightarrow$) Let $\mu_i u_i \geq (s_{ij} - r_{0i})^2 x_{ij}$ for facility i and demand point j. (i) $x_{ij} = 1$, $\mu_i u_i \geq (s_{ij} - r_{0i})^2 \Longrightarrow \mu_i u_i \geq (s_{ij}x_{ij} - r_{0i})^2$. (ii) $x_{ij} = 0$, $\mu_i u_i \geq 0 \Longrightarrow r_{0i} + \sqrt{\mu_i u_i} \geq 0$.

($\Leftarrow$) Let $\mu_i u_i \geq (s_{ij}x_{ij} - r_{0i})^2$. $(s_{ij}x_{ij} - r_{0i})^2 \geq x_{ij}(s_{ij}^2 x_{ij} + r_{0i}^2 - 2s_{ij}x_{ij}r_{0i})$ ($\because x_{ij} \leq 1$). In both cases of $x_{ij} \in \mathbb{B}$, $\mu_i u_i \geq (s_{ij} - r_{0i})^2 x_{ij}$ holds. $\square$

Based on Proposition 1, Constraint (4.5′) can be substituted for Constraint (4.5):

$$u_i \geq CV_{ij}x_{ij}, \qquad \forall i \in I, \forall j \in J. \tag{4.5′}$$

4.3.4 Linear reformulation of standard formulation

The standard formulation with nonlinear constraint can be reformulated into an equivalent linear formulation using the dual of the inner optimization problem [18]. The reformulation is provided as follows:

[Reformulated standard formulation]

$$\min \quad (4.1)$$

$$\text{s.t.} \quad (4.2), (4.3), (4.5'), (4.6), (4.7) - (4.9)$$

$$u_i \geq \sum_{j \in J} d_j^0 x_{ij} + \alpha_i \Gamma_i + \sum_{j \in J} \beta_{ij}, \qquad \forall i \in I \tag{4.10}$$

$$\alpha_i + \beta_{ij} \geq \hat{d}_j x_{ij}, \qquad \forall i \in I, \forall j \in J \tag{4.11}$$

$$\alpha_i \geq 0, \qquad \forall i \in I \tag{4.12}$$

$$\beta_{ij} \geq 0, \qquad \forall i \in I, \forall j \in J. \tag{4.13}$$

The decision variable β_{ij} is an auxiliary variable used for the linearization of the inner optimization problem of Constraint (4.4′). α_i is the dual variable of the linearized inner optimization problem. Constraint (4.10) defines the capacity constraint of a facility, which uses a protection function to linearize Constraint (4.4′). Constraint (4.11) is the dual of the linearized inner optimization problem. For detailed information on the reformulation procedure, one can refer to Bertsimas and Sim [18]. Note that the reformulated standard formulation (RSF) consists of linear constraints with binary, integer, and continuous decision variables. The problems in the RSF can be solved with commercial optimization solvers (e.g., Cplex, Xpress, and Gurobi). However, the commercial solvers that utilize LP relaxation-based branch-and-cut algorithms have difficulty solving the large-sized problems in the RSF within

a short computation time, even though the solvers can handle relatively large-sized nominal problems within reasonable computation times. This is because the combination of the integer and continuous decision variable generally causes the highly fractional solution of the LP relaxation and have a weak LP relaxation bound [7].

4.4 Solution algorithms

4.4.1 An extended formulation of the ULAP

The solution of the ULAP consists of individual decisions about each facility, which constructs a set-partitioning structure. Based on the Dantzig-Wolfe decomposition, one can reformulate the decisions of the SF with *column-wise* decisions. Each column in the extended formulation defines a set of demand points covered by a facility and the minimum number of UAVs required for the allocation. In other words, based on each facility, several feasible allocation of the set of demand points are given in advance, along with the required number of UAVs. Ω_i is a set of feasible columns for a facility, i. A column is defined as a combination of the subset of the demand points that can be allocated to a facility and the number of UAVs required to cover the demand points. The parameters and the decision variables of the extended formulation are presented as follows:

Parameters

c_{ik} cost associated to column k of facility i.

w_{ij}^k indicate whether demand point j is covered by facility i in column k.

u_i^k number of UAVs assigned to facility i in column k.

Decision variables

$$z_{ik} = \begin{cases} 1, & \text{if column } k \text{ is used by facility } i. \quad \forall i \in I, \\ 0, & \text{otherwise.} \quad \forall k \in \Omega_i \end{cases}$$

The cost of each column is defined as $c_{ik} := f_i + p_i u_i^k$. The extended formulation (EF) model of UEMS is represented in the following integer program:

[Extended formulation]

$$\min \quad \sum_{i \in I} \sum_{k \in \Omega_i} c_{ik} z_{ik} \tag{4.14}$$

$$\text{s.t.} \quad \sum_{i \in I} \sum_{k \in \Omega_i} w_{ij}^k z_{ik} \geq 1 \qquad \forall j \in J \tag{4.15}$$

$$\sum_{k \in \Omega_i} z_{ik} \geq 1 \qquad \forall i \in I \tag{4.16}$$

$$z_{ik} \in \mathbb{B} \qquad \forall i \in I, \forall k \in \Omega_i \tag{4.17}$$

The extended formulation only remains the set-partitioning structure, while the capacity-related and coverage-distance-related constraints are considered implicitly in the column. Let us call the LP relaxation of the extended formulation the *master problem*. The LP dual of the master problem can be viewed as the Lagrangian dual. Thus, the master problem can provide the Lagrangian dual, which is better than the LP bound of the standard formulation. By Minkowski's theorem, every solution of the compact formulation can be represented in the extended formulation. If Ω_i contains every feasible column for every facility i, then the solution set of the master problem defines the convex hull of the ULAP. However, this requires an exponential number of columns. To avoid maintaining a very large number of variables, the

column generation (CG) technique can be implemented to solve the Lagrangian dual. The CG iterates between the restricted master linear problem (RMLP) and the pricing subproblem while generating new variables that might improve the current solution. Let π_j and σ_i be dual prices associated with Constraints (4.15) and (4.16). The pricing subproblem can be defined for each facility i:

[Pricing subproblem]

$$\min \quad f_i + p_i u - \sum_{j \in J} \pi_j x_j + \sigma_i \tag{4.18}$$

$$\text{s.t.} \quad u \geq \sum_{j \in J} d_j^0 x_j + \max_{N \subseteq J, |N| = \Gamma_i} \sum_{j \in N} \hat{d}_j x_j, \tag{4.19}$$

$$u \geq C V_{ij} x_j, \qquad \forall j \in J \tag{4.20}$$

$$u \leq n_0 \tag{4.21}$$

$$x_j \in \mathbb{B}, \qquad \forall j \in J \tag{4.22}$$

$$u \in \mathcal{Z}_+. \tag{4.23}$$

The pricing subproblem of ULAP can be defined as a *robust integer knapsack problem*. Pochet and Wolsey [86] and Ceria et al. [25] defined the integer knapsack constraint as a knapsack constraint, which is including integer decision variables in the capacity constraint. The integer decision variable acts as a supplementary capacity that can be purchased additionally from the original capacity. Constraint (4.19) is nonlinear and can be reformulated as in the RSF. If Constraint (4.19) is reformulated with the robust counterpart, the pricing subproblem gets a characteristic of a robust mixed-integer linear knapsack problem because of the integer decision variable u_i and the continuous variables α_i and β_{ij}. Constraint (4.20) hunts

down the demand points that cannot be covered by the given coverage distance, and zeroes out the decision variable, x_j, of the demand points. On the other hand, when the number of UAVs assigned to a facility is fixed, it becomes a robust 0-1 knapsack problem. Therefore, the pricing subproblem can be solved by a decomposed approach that solves multiple 0-1 robust knapsack problems. The number of the decomposed 0-1 robust knapsack problems equals to n_i, the maximum number of UAVs to be operated at the UEMS facility i.

4.4.2 Branching strategy

The column generation iterates between the restricted master problem and the pricing subproblem. The solution of the master problem can be translated into the solution space of the original (standard formulation) variable. However, because the master problem is the LP relaxation of the extended formulation, the translated solution is not necessarily an integer when the column generation process is solved to the optimal. The branch-and-price algorithm applies the CG to solve each node in the branch-and-bound procedure, and can execute branching when the solution of the CG is fractional.

The branching in the B&P algorithm can be executed with various strategies [8]. First, the branching can be executed over the variables of the extended problem. However, it is generally difficult to formulate the branching decision explicitly in the variables of the pricing problem, and can complicate the solution algorithm. Also, binary fixing of one among many columns provides an unbalanced branch-and-bound tree, which weakens the effect of the branching.

In the second strategy, branching over the original variables of the standard

formulation can be used. In the ULAP, for a facility i, if the decision variable x_{ij} is decided for every demand point j, then the minimum value of u_i can be calculated subsequently:

$$u_i = \max\left\{\left\lceil \sum_{j\in J} x_{ij}\tilde{d}_j \right\rceil, \left\lceil [(x_{ij}s_{ij} - r_i^0)^+]^2/\mu \right\rceil\right\}.$$

Thus, the branching can be defined based on the decision variable x_{ij}. When the CG provides a fractional solution of the original variable x_{ij}, the branching creates two children nodes separated by the allocation between a demand point and a facility. The second strategy shows a more balanced branch-and-bound tree, so it is commonly used for various applications (e.g., generalized assignment problems). One strength of the branching on original variables is that the branching decision does not complicate the pricing subproblem. In the robust knapsack problem, fixing the allocation of a demand point can be applied by adjusting the remaining capacity of the facility. Thus, the branching only changes the parameters of the pricing subproblem while maintaining the structure of the problem. However, in the ULAP, binary fixing of one decision variable x_{ij} still provides an unbalanced branching and has a small impact. In the computational experiments in Section 4.5, we observed the inefficiency of the branching on original variables.

The third strategy is the Ryan-Foster [89] branching rule. When the master problem has a set-partitioning-like characteristic over a pure binary subproblem, the branching decision can be made based on the coexistence of two elements [107]. We applied the Ryan-Foster branching rule, which showed the best performance based on the model of the balanced branching tree. However, if the problem does not originally include conflict constraints, unlike edge coloring and bin packing with conflict [90],

the implementation of the Ryan-Foster branching rule may change the structure of the pricing subproblem. Thus, in order to apply the Ryan-Foster branching rule to the ULAP, we redefined the pricing subproblem, especially in order to implement the special-purpose solver for the subproblem. In the Ryan-Foster branching rule, the fractional solution of the restricted master problem denotes the (fractional) employment of a feasible column. Based on the fractional solution of the CG, the degree of the coexistence of a pair of demand points, v_{j_1,j_2}, can be measured. If a column, $k \in \Omega_i$, includes both demand points j_1 and j_2, then the optimal solution of the column, z^*_{ik}, can be added to measure the degree of the coexistence of the pair of demand points j_1 and j_2. One can calculate it for every pair of demand points in the same way: $v_{j_1,j_2} := \sum_{k \in \Omega_i, w^k_{i,j_1} = w^k_{i,j_2} = 1} z^*_{ik}$. When the degree of coexistence is nearest to 0.5, then the pair of demand points is chosen for the branching.

4.4.3 Robust disjunctively constrained integer knapsack problem

As mentioned in Section 4.4, the pricing subproblem of the ULAP is related to the robust integer knapsack problem. Under the determined number of UAVs assigned to a facility, the coverage distance is fixed and can be used to define the feasible set of demand points based on Constraint (4.20). The subproblem with the remaining feasible demand points forms a robust 0-1 knapsack problem.

The Ryan-Foster branching provides two children, one forcing and the other forbidding the coexistence of two demand points. The former is called *same-child*, and is easy to be considered in the special-purpose solver by introducing an auxiliary demand point merging two demand points. However, the latter *differ-child* destroys the special structure of the knapsack problem. The conflict, or disjunctive

constraint, in the knapsack problem is notorious for its simple shape and difficulty. The disjunctively constrained knapsack problem (DCKP) was defined by Yamada et al. [113]. However, the DCKP drew more attention because of its equivalence [90] to the pricing subproblem of the bin packing problem with conflict (BPPC). The DCKP was further investigated by Elhedhli et al. [43] and Sadykov and Vanderbeck [90] while solving the BPPC.

In Section 4.4.5, the generic branching scheme [105] was used to implement the Ryan-Foster branching in the ULAP and consider the following arbitrary conflict. Based on the conflict relation between demand points defined by the branching, we defined the feasible set and solved the pricing subproblem individually.

As a matter of convenience, we call the subproblem of the ULAP a robust disjunctively constrained integer knapsack problem (RDCIKP). In this chapter, two solution approaches are proposed to solve the RDCIKP. In Section 4.4.4, we find linear reformulation of the RDCIKP and solve with the MILP solver. In Section 4.4.5, the RDCIKP is decomposed into multiple 0-1 knapsack problems and solved with the dynamic programming algorithm. In the computational experiment, we used a hybrid algorithm using both approaches alternately.

4.4.4 MILP reformulation approach

By using the reformulation of Bertsimas and Sim [18], we derived the linear reformulation of the RDCIKP for a facility i:

[Linear reformulation of RDCIKP]

$$\begin{aligned} \min \quad & (4.18) \\ \text{s.t.} \quad & (4.20) - (4.23) \end{aligned}$$

$$u \geq \sum_{j \in J} d_j^0 x_j + \alpha \Gamma_i + \sum_{j \in J} \beta_j, \tag{4.24}$$

$$\alpha + \beta_j \geq \hat{d}_j x_j, \qquad \forall j \in J \tag{4.25}$$

$$\alpha \geq 0, \tag{4.26}$$

$$\beta_j \geq 0, \qquad \forall j \in J. \tag{4.27}$$

When branching happens in a node of the branching tree, if the demand points j_1 and j_2 are chosen, then an additional constraint, $x_{j_1} = x_{j_2}$, is added to every subproblem of the same-child node, regardless of the facility. On the other hand, a more complicated constraint, $x_{j_1} + x_{j_2} \leq 1$, is added to every subproblem of the differ-child node. As in the reformulated standard formulation, the nonlinear capacity constraint is linearized using the dual variable of the inner optimization problem. Thus, the model can be solved with the MILP solver. The formulation consists of the binary, integer, and continuous decision variables, which give rise to the weak LP bound. In the early stages of the CG process, linear reformulation of the RDCIKP can be solved within a short computation time with a commercial solver. However, as the CG progresses and the dual bound converges to the optimal dual solution of the master problem, it gets difficult and takes time to solve the pricing subproblem. Thus, a decomposed DP approach is designed to use a special-purpose solver to solve the RDCIKP, as presented in the next section.

4.4.5 Decomposed DP approach

The 0-1 knapsack problem is one of the most studied problems and is well known to be solved by dynamic programming in pseudo-polynomial time. To take advantage of

the knowledge of the knapsack problem, we propose a decomposed-based approach in this research. From the ideas of Bertsimas and Sim [17], the RDCIKP is decomposed into nominal 0-1 knapsack problems, based on the combination of the feasible sets.

Bertsimas and Sim [17] showed that the robust 0-1 knapsack problem could be solved by solving $|J| + 1$ nominal 0-1 knapsack problems. Lee et al. [72] reduced the number of nominal problems into $|J| - \Gamma + 1$. On the other hand, as shown in Section 4.4.3, if the number of UAVs assigned to a facility is fixed, then the RDCIKP becomes a robust disjunctively constrained 0-1 knapsack problem. Thus, the RDCIKP can be decomposed by the number of UAVs, u. For the disjunctive constraint, Pferschy and Schauer [84] proposed a pseudo-polynomial algorithm for the DCKP with chordal conflict graphs. However, the ULAP consists of the arbitrary conflict relation, which requires the enumeration of the feasible set. Let us consider the feasible set of a RDCIKP of facility i defined as:

$$
\begin{aligned}
S =& \Big\{ x \in \mathbb{B}^m, u \in \mathcal{Z}_+ \Big| u \geq \sum_{j \in J} d_j^0 x_j + \max_{M \subseteq J, |M| = \Gamma_i} \sum_{j \in N} \hat{d}_j x_j, u \leq n_0, \\
& \qquad\qquad u \geq CV_{ij} x_j, \text{ for all } j \in J, x \in \mathcal{B} \Big\} \\
=& \Big\{ x \in \mathbb{B}^m, u \in \mathcal{Z}_+ \Big| u \geq \sum_{j \in J} d_j^0 x_j + \sum_{j \in J'} \hat{d}_j x_j, u \geq CV_{ij} x_j, u \leq n_0, \\
& \qquad\qquad \text{for all } j \in J, \text{ for all } J' \subseteq J \text{ with } |J'| = \Gamma_i, x \in \mathcal{B} \Big\}
\end{aligned}
$$

where $\mathcal{B}$ is the family of all the subsets of demand points that are not in conflict. In practical implementation, we developed independent sets of the given conflict graph with enumeration. The demand points, without any disjunctive constraint, were then added to each independent set, which formed a feasible set, $\mathcal{B}$.

Following the notations of Lee et al. [72], let us define sets $L = \{\Gamma_i, \Gamma_i+1, ..., m-1, m+1\}$ and $S_{ul} = \{x \in \mathbb{B}^m | u - \Gamma_i \hat{d}_l \geq \sum_{j\in J} d_j^0 x_j + \sum_{j\in J_l}(\hat{d}_j - \hat{d}_l)x_j, u \geq CV_{ij}x_j,$ for all $j \in J\}$, where $J^+ = J \cup \{m+1\}$, $l \in J^+$, and $J_l = \{j \in J^+ | j \leq l\}$. By the following proposition, the RDCIKP can be solved by solving at most $2^E n_i(|J| - \Gamma_i + 1)$ nominal 0-1 knapsack problems, where E is the number of the disjunctive constraints of the RDCIKP.

Proposition 2. *The RDCIKP*

$$Z^* = \max\Big\{\sum_{j\in J} \pi_j x_j - f_i - \sigma_i - p_i u \Big| (x,u) \in S, x \in \mathcal{B}\Big\}$$

can be solved by solving at most $n_i(m - \Gamma_i + 1)$ nominal disjunctively constrained 0-1 knapsack problems

$$Z^*_{ul} = \max\Big\{\sum_{j\in J} \pi_j x_j - p_i u \Big| (x,u) \in S_{ul}, x \in \mathcal{B}\Big\},$$

$$\textit{for all } u \in \{0, ..., n_i\}, l \in L$$

Proof. Note that f_i and σ_i are given parameters. For every $u \in \{0, \ldots, n_0\}$, let us define a set $S_u = \{x \in \mathbb{B}^m | u \geq \sum_{j\in J} d_j^0 x_j + \max_{M\subseteq J, |M|=\Gamma_i} \sum_{j\in N} \hat{d}_j x_j, u \geq CV_{ij}x_j,$ for all $j \in J\}$ which is a subset of S. Because $S = \bigcup_{u=0}^{n_i} S_u$, Z^* can be solved by solving at most n_i robust disjunctively constrained 0-1 knapsack problems. By Lee at al. [72], each robust disjunctively constrained 0-1 knapsack problems can be solved by solving at most $|J| - \Gamma_i + 1$ nominal disjunctively constrained 0-1 knapsack problem. □

The feasible set, $\mathcal{B}$, used in the proposition is defined based on the independent

set, which has cardinality at most 2^E. By the rule of product of the nested loop, the RDCIKP can be solved in $O(2^E n_i^2 |J|^2)$.

4.4.6 Restricted master heuristic

Despite the better LP bound of the CG and the fast computation speed of the pricing subproblem, the B&P algorithm can take a long enumeration of branching on the large-sized problem, resulting in a long computation time. To solve the large-sized problem in a feasible computation time, we implemented the *restricted master heuristic (RMH)*, as in the UVCP. It is not guaranteed that all the values required are found before starting to solve the master integer problem, so the optimal solution of the heuristic can only be used to find the primal bound of the ULAP. Note that the RMH can be initiated in any stage of the B&P algorithm, especially before the termination of the root node CG.

Although there are advanced heuristics, including the diving heuristic [91], developed for the B&P algorithm, they are challenging to implement in the ULAP. Note that the best-first search strategy is implemented in the B&P algorithm to maximize the advantage of the Lagrangian dual bound. The diving heuristics are developed based on the depth-first search, which complicates the utilization of the heuristics in the ULAP. One alternative is the relax-and-fix algorithm on the original variables. It also has difficulty with the highly fractional solution of the LP relaxation induced by the robust counterpart and the following continuous decision variables. As mentioned in Section 4.4.2, fixing one decision variable, x_{ij}, did not provide a dramatic effect on the residual problem. Also, there is a feasibility issue using the relax-and-fix algorithm in the ULAP because of the capacity constraint.

The RMH can be implemented inside of the B&P algorithm to provide the primal bound. Furthermore, if there is a substantial restriction on the computation time, the heuristic can be used to provide a feasible solution, which is presented in Section 4.5.

4.5 Computational experiments

Computational experiments were conducted to compare the performance of the proposed solution algorithms. The models were developed in FICO Xpress 8.5 and solved with Xpress-Optimizer 33.01.02. Experiments were performed with an AMD Ryzen ™ 7 2700X 8-Core CPU at 3.70GHz and 32GB of RAM running on a Windows 10 64-bit operating system.

4.5.1 Datasets used in the experiments

Small and large-sized datasets were randomly generated for computational experiments, using the simple plant location problem on a Euclidean plane, with benchmark data from the Benchmark Library [4]. Two small-sized and three large-sized problem classes were tested, and 10 instances are generated for each problem class. The demand points were distributed randomly on the interior of a given size of a square on the Euclidean plane. In order to achieve the realistic distribution of candidate locations of UEMS facilities, the Euclidean plane was divided into a lattice. The plane was divided into the largest number of squares, which was smaller than the number of candidate locations of the UEMS facilities $|I|$. For example, if $|I| = 20$, the plane was divided into 16 cells. The candidate locations were first distributed on each cell. After filling every cell, the rest of the candidate locations (four candidate

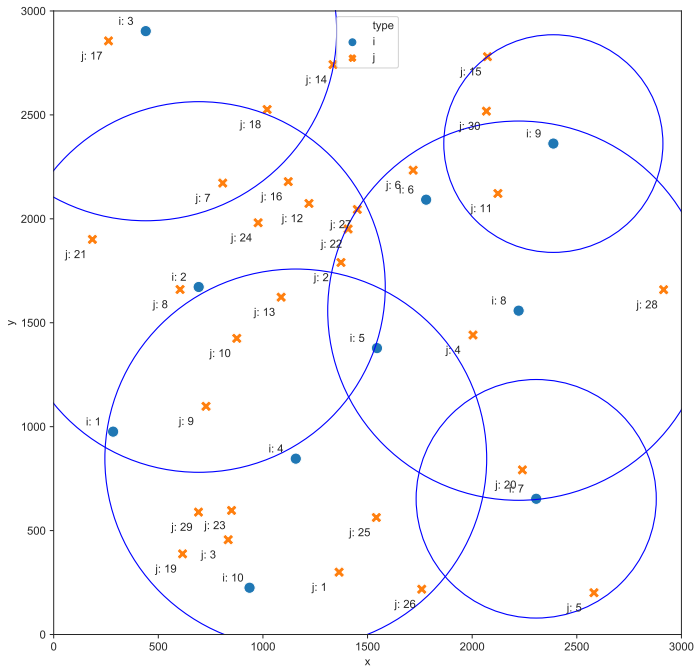
locations in this example) were distributed without any restriction. Figure 4.2a and Figure 4.2b show solutions of a small and large-sized problem, respectively.

For each problem class, the maximum number of UAVs operated at the UEMS facility, the minimum ensured coverage distance, and the variable coverage distance ratio per UAV were determined for the realistic UEMS system in regard to both capacity and coverage distance, both of which can affect the number of UAVs assigned to a facility. The parameters related to the number of UAVs, capacities, and coverage distances are presented in Table 4.2. In the table, NI and NJ represent the number of candidate facilities and the demand points, respectively. $Slim$ refers to the size of the plane.

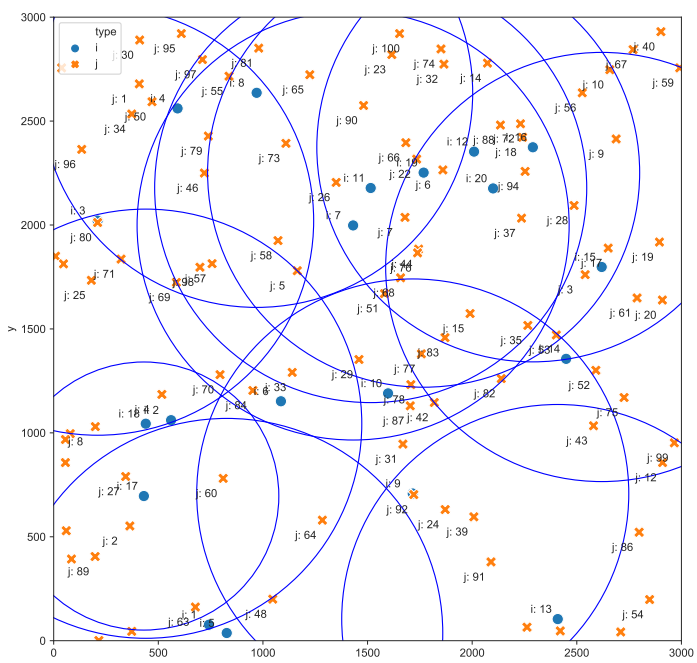
The demand for each demand point was randomly generated based on uniform distribution. The nominal value of the demand d_j^0 and its maximum perturbation $\hat{d}_j$ are generated following U[0, 3] and U[0, $d_j^0/1.3$], respectively. The opening cost and operation (assignment) cost per unit UAV were generated based on Shavarani et al. [97], which followed U[300,000, 400,000] and U[30,000, 40,000], respectively.

Table 4.2: Parameters of the problem classes.

Problem Class	NI	NJ	Γ	n_i	r_{0i}	μ_i	$Slim$
C11	10	10	3	20	500	200,000	5,000
C13	10	30	4	15	300	25,000	3,000
C23	20	30	4	12	300	25,000	3,000
C210	20	100	10	20	1200	20,000	3,000
C320	30	200	10	20	500	200,000	5,000



(a) RSF solution of C130.



(b) RMH solution of C2100.

Figure 4.2: Example of the solution.

4.5.2 Algorithmic performances

First, we compared the algorithmic performances of the two formulations, the RSF and the EF. The tightness of the LP relaxation bounds and the induced lower bounds were compared, along with the optimality of the primal bounds. Second, a further analysis of the B&P algorithm was conducted. Third, the performances of the restricted master heuristics were analyzed.

For the first analysis, the computation speed was compared based on the computation time and the final gap between the best value of the feasible solution (BFS) and the best lower bound (BB). The optimality of the BFS was compared. The strength of the formulation was compared based on the tightness of the LP relaxation and the final BB. In the computation, the maximum time limit was set to 3,600 seconds. In Tables 4.3 – 4.5, the results are summarized by the average of 10 instances of each problem class. The columns in these tables are defined as follows:

- *#solved*: the number of the instances in the problem class, solved by each algorithm.
- *Time*: the average of the computation time. For the problems not solved within the computation time limit, the limit was used as the computation time while calculating the average, and was marked with an asterisk (*).
- Gap_L: the average of the gap between the primal bound (the best feasible solution) and the best lower bound. Gap_L is used to evaluate the algorithm's convergence speed, especially for the problems not solved to optimal within the time limit.

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

- Rat_{LP}: the average of the ratio between the LP relaxation bounds of the reformulated standard formulation and the extended formulation.

$$Rat_{LP} = \frac{\text{(LP bound of the EF)}}{\text{(LP bound of the RSF)}} \times 100\%$$

The pricing subproblem is identical to the Lagrangian relaxation subproblem, and each iteration of the column generation provides a Lagrangian dual bound, which can be used as a lower bound of the RMLP [107]. Thus, in the CG of the root node, the dual bound of a pricing subproblem can be used as a lower bound of the problem, regardless of the termination of the CG. On the other hand, in the B&P algorithm, we implemented the best-first search (BS) for the search strategy. Because the lower bound of the head node of the active node queue always has the *worst* lower bound in the BS, it can be used as a lower bound of the problem. For the problems that the B&P algorithm has not finished within the time limit, the lower bound calculated by the BS was used for the comparison in Table 4.4. In the same table, if the root node CG was not finished, the lower bound of the RMLP was used as the LP relaxation bound and marked with an asterisk (*).

In the small-sized problems, the commercial solver's root cutting and heuristic techniques showed powerful performances while solving the RSF. Even the integer feasible solutions provided by the root cutting and heuristics were optimal solutions in several small-sized problems. The B&P algorithm could solve the problem within a short computation time without both a heuristic and a valid inequality, due to the advantage of the LP relaxation bound. Even though the B&P algorithm generally could not solve the small-sized problems faster than the RSF solved by the com-

mercial solver, it could solve nine instances of the problem class C13 in the average computation time of 0.75 seconds and solve the other instance in 78.49 seconds. In the problem class C23, two instances were not solved within the time limit, but the rest of the instances were solved in the average computation time of 318.42 seconds. The average of the overall instances of each problem class is calculated in Table 4.3.

In the large-sized problems, the root node CG of the problem class C210 was solved within the time limitation, but the overall B&P was not finished as well as the RSF. If an algorithm failed to solve every instance in a problem class, the average computation time was marked to be the time limit 3,600. The size of the problem class C320 was too big to be solved by any algorithm. However, the progress of the optimization, including the CG and the RMH algorithm, was compared later.

Gap_L compares the convergence between the BFS and the BB of algorithms. In the problem classes C210 and C320 of the B&P algorithm, the BFS was not provided, so it was impossible to calculate the Gap_L. As mentioned above, the commercial solver could provide effective feasible solutions of the RSF with root cutting and heuristics. When implemented in real-world cases, those techniques could be utilized in the B&P algorithm by providing the feasible columns and the primal bounds. In this computational experiment, we investigated the advantages of the EF in the BB perspective. Thus, the LP bound and the lower bound provided by the following branching are compared in Table 4.4, without consideration of additional techniques, such as valid inequalities and heuristics.

As mentioned in Section 4.4.1, the LP bound of the EF was always the same or better than the LP bound of the RSF. Indeed, as illustrated in Table 4.4, the LP relaxation bound of the EF was better than the RSF from 1.7% to 81.1%, and

the overall average of the Rat_{LP} was 138.8%. In the problem class C320, the LP bound has the same value of the BB because the root node column generation was not solved to the optimal. It was expected that the gap of the LP bound between the RSF and the EF would increase when the CG was finished. As the optimization process continued, the *Gap* of the BB was marked as zero if both algorithms found the optimal solution. There was a minus gap of the lower bound in the problem class C23, which took place because of the two unfinished instances of the B&P algorithm. As the size of the problem increased, the gap of the lower bound between the RSF and the B&P algorithms deepened.

In the optimality perspective of the small-sized problems, in Table 4.5 there is no gap between the two algorithms because both algorithms found the optimal solution for every problem. In the large-sized problems, the B&P algorithm could not find feasible integer solutions. In this research, the B&P algorithm employed the best-first search to utilize the advantage of the lower bound. The depth of the branching tree does not deepen fast in the BS, so it is difficult to find the primal solution. Although the primal heuristic, such as the RMH, can help to find the feasible solution, it was not used in this experiment to examine the primary performance of the EF and the B&P themselves. The performance of the RMH will be investigated in the latter part of this paper.

For the second analysis, the nodes and the columns generated in the B&P algorithm are summarized in Table 4.6. The effectiveness of the EF and the column generation algorithm are analyzed in Table 4.7. The columns in the tables are defined as follows:

- *Cols*: the average number of columns generated in the CG and the B&P algo-

Table 4.3: Results related to the computation speed.

Problem Class	# *solved*			*Time* (s)			Gap_L (%)	
	RSF	CG	B&P	RSF	CG	B&P	RSF	B&P
C11	10	10	10	0.06	0.05	0.44	0.0	0.0
C13	10	10	10	0.25	0.47	8.52	0.0	0.0
C23	10	10	8	1.15	0.79	974.36*	0.0	0.6
C210	0	10	0	3,600*	1,294.63	3,600*	7.1	-
C320	0	0	0	3,600*	3,600*	3,600*	34.3	-

$*$: There are instances not solved within the time limit.
$-$: There is no instance that provided a BFS within the time limit.

Table 4.4: Comparison of LP relaxation.

Problem Class	*LP relaxation bounds*			*Lower bounds*		
	RSF	EF (CG)	Rat_{LP} (%)	RSF	B&P	*Gap* (%)
C11	770,575	993,507	127.7	1,029,704	1,029,704	0.0
C13	1,878,139	3,082,336	164.4	3,125,399	3,125,399	0.0
C23	2,108,738	3,069,304	146.4	3,202,811	3,184,722	-0.6
C210	4,968,062	6,269,128	126.1	6,037,631	6,269,589	3.8
C320	9,252,454	11,975,240*	129.4	9,894,785	11,975,240*	21.0

$*$: There are instances with the unfinished root node CG.

Table 4.5: Results related to the optimality.

Problem Class	RSF	B&P	
	BFS	BFS	*Gap* (%)
C11	1,029,705	1,029,704	0.0
C13	3,125,400	3,125,399	0.0
C23	3,202,813	3,202,813	0.0
C210	6,507,051	-	-
C320	15,484,665	-	-

$-$: No BFS is found within the time limit.

rithms. If a problem class contains instances not solved within the computation time limit, the number of the columns generated up until the time limit was

used to calculate the average and marked with an asterisk (*).

- $Nodes$: the average number of nodes generated in the B&P algorithm. For problem classes with problems not solved within the computation time limit, the number of the nodes generated up until the time limit were used and marked with an asterisk (*).
- $\#\ CG_{opt}$: the average number of problems that the root node CG provided integer solutions.
- Gap_{int}: the average of the integrality gap between the optimal solution and the LP bound of the EF provided by the CG algorithm. For the problems by which we do not have optimal solutions, the best feasible solutions found from any algorithm are used and marked with an asterisk (*). If the root node CG is not finished, the lower bound of the RMLP is used as the LP relaxation bound, and the problem class is marked with a dagger symbol (†).

$$Gap_{int} = \frac{\text{(Optimal solution)}}{\text{(LP bound of the EF)}} \times 100\%$$

- $\#\ Gap_{int}$: the number of problems that have values of Gap_{int} less than a certain criteria. That is, the value of the Gap_{int} should be less than 100.5% or 105% to be counted in the columns "$< 0.5\%$" or "$< 5\%$," respectively.

In Table 4.6, it is shown that as the size of the problem increased, the number of columns and nodes required followed. Note that the optimization process of the large-sized problems was not fully terminated, and the additional columns and nodes would be generated afterward. In the RSF model, C210 and C320 consisted of 4,060

and 12,090 variables, respectively. Compared to the variables generated in B&P algorithms, the RSF models tended to be larger in sizes.

In Table 4.7, the performance of both the EF and the CG algorithm was analyzed from the integrality gap point-of-view. In the small-sized problems, the CG algorithm provided an integer optimal solution for 13 out of 30 instances, which did not require the additional branching process. The integrality gap is the ratio between the optimal integer solution and the LP relaxation bound provided by the CG algorithm. The smaller integrality gap indicates the advantage of the EF. In the problem classes C11 and C13, 15 out of 20 instances had an integrality gap of less than 0.5%. In the larger problems, the integrality gap increased. However, it should be considered that in these problems, the BFS was used instead of the optimal solution because the large-sized problems were not solved to the optimal. Besides, for the class C320, the LP bound would also be improved when the CG is solved to the optimal.

Table 4.6: Results related to the CG and B&P algorithm.

Problem Class	*Cols*		*Nodes*
	CG	B&P	B&P
C11	75.9	131.8	5.9
C13	173.6	271.2	14.0
C23	252.1	1,089.5*	351.8*
C210	1,682.5	2,214.2*	14.0*
C320	4,902.1*	4,902.1*	-*

$*$: There are instances not solved within the time limit.
$-$: There is no instance with the finished root node CG.

The performances of the restricted master heuristics were tested in the third analysis. The RMH was proposed to solve the large-sized problem in a real-world situation in a time-efficient manner. As mentioned in Section 4.4.6, the RMH could

Table 4.7: Results related to the EF and the CG algorithm.

Problem Class	# CG_{opt}	Gap_{int} (%)	# Gap_{int}	
			$< 0.5\%$	$< 5\%$
C11	7	104.7	8	8
C13	6	101.5	7	9
C23	0	104.4	1	7
C210	0	103.8*	0*	8*
C320	-	122.2*†	0*†	0*†

$*$: There are instances that used the BFS to calculate the integrality gap.
† : Lower bound of the RMLP is used to calculate the integrality gap.

be initiated at any stage of the B&P algorithm. The feasibility of the RMH was secured based on the initial columns, which were generated according to every possible pairing of a demand point and a facility. In the computational experiment, we compared three initiation points of the RMH. The first point was the well-known price-and-branch, where the RMH started after solving the root node CG. The second and third points were defined based on the computation time. In this research, because the computation time limit is set to 3,600 seconds, we tested to start the RMH 1,200 and 2,400 seconds after the beginning of the B&P algorithm. Three initiation points are named in Tables 4.8 and 4.9 as "RT," "1200," and "2400," respectively. In Table 4.9, column "# (*same*, *better*) *BFS*" represents the number of problems that the RMH algorithm provided the same or better BFS than the RSF. Note that in the small-sized problem classes in Table 4.9, the BFS of the RSF is the optimal solution.

In the RMH with the second and third starting points, there were problems solved by the B&P before the RMH was initiated. Every instance in problem classes

C11 and C13, and eight instances in problem class C23 were solved before the RMH was started at 1200. In class C23, because the CG generated around 250 columns, the RMH with the starting point RT (RMH_RT) solved the set partition problem with 250 decision variables. Compared to the B&P algorithm with an average computation time of 974.36 seconds, the RMH_RT provided the solution faster with the average computation time of 0.86 seconds. Even for the RMH, problem class C320 was hard to solve within the time limitation of 3,600 seconds, however. Nevertheless, the RMH had relatively smaller Gap_L than did the RSF, where the CG provided a better LP bound, and the heuristic provided a decent feasible solution.

Table 4.9 compared the optimality of the RMH to the RSF. In the RMH with the starting point of 1,200 seconds (RMH_1200), the optimal solution was found for every instance of the small-sized dataset. For the large-sized problem class, C210 and C320 had 4.5 and 8.5 percent of an average optimality gap, respectively. Furthermore, in problem class C320, RMH_1200 provided better BFS than RSF in four instances. As mentioned above, the size of the problem was smaller in the RMH than the RSF, and the RMH starts with the better LP relaxation bounds. However, if the starting time of the RMH was overdue, the RMH started with too many columns, and it became hard to find the feasible solution for the RMH. In an instance of problem class C320, RMH_2400 could not provide a feasible solution. The average of Gap_L and Gap were calculated without the instance, and the problem class was marked with a dagger symbol (†). The effective starting point of the RMH would be affected by each problem situation, and a further investigation is required when implemented in the applications.

To summarize, the EF provided a strong LP relaxation bound. The LP relax-

Table 4.8: Results of the RMH related to the computation speed.

Problem Class	# *solved*			*Time* (*s*)			Gap_L (%)		
	RT	1200	2400	RT	1200	2400	RT	1200	2400
C11	10	10	10	0.08	0.44	0.44	0.0	0.0	0.0
C13	10	10	10	0.79	8.52	8.52	0.0	0.0	0.0
C23	10	10	10	0.86	370.29	735.22	0.0	0.0	0.0
C210	7	8	7	2,169.55*	2,319.65*	2,998.55*	1.2	0.9	2.2
C320	-	0	0	-	3,600*	3,600*		19.2	27.1†

$*$: There are instances not solved within the time limit.
† : There is an instance that a feasible solution is not found within the time limit.

Table 4.9: Results of the RMH related to the optimality.

Problem Class	*Gap* (%)			# (*same*, *better*) *BFS*		
	RT	1200	2400	RT	1200	2400
C11	0.9	0.0	0.0	(7, 0)	(10, 0)	(10, 0)
C13	0.3	0.0	0.0	(8, 0)	(10, 0)	(10, 0)
C23	1.6	0.0	0.0	(2, 0)	(10, 0)	(10, 0)
C210	4.2	4.5	3.2	(0, 0)	(0, 0)	(0, 1)
C320	-	8.5	14.9†	-	(0, 4)	(0, 3)

† : There is an instance that a feasible solution is not found within the time limit.

ation bound of the EF had a competent integrality gap, so the B&P algorithm was expected to show a good performance in the lower bound point-of-view. In the large-scale problems, it was observed that the RMH could provide time-efficient solutions. The RMH could be implemented in the B&P algorithm to find primal bounds used in the branching. The other heuristic algorithms could be used alongside of the RMH and provide initial columns of B&P algorithms for solving real-life problems. It was observed that the algorithms compared in the computational experiments could not solve the large-scale problems of C210 and C320 to the optimal within 3,600 sec-

onds. When implementing the ULAP to the real large-scale applications, one can implement existing techniques of demand point aggregation [60] or divide the overall problem for the applicable scales.

4.5.3 Analysis of the branching strategy and the solution approach of the pricing subproblem

First, we compare two branching strategies introduced in Section 4.4.2. In the computational experiments, the Ryan-Foster branching rule was used for the balanced and efficient branching trees. To compare the two branching strategies, we selected the instances that required branching in the B&P algorithm. In Tables 4.10 and 4.11, the Ryan-Foster branching rule and the branching over the original variable are named as "RF." and "Org.," respectively. Table 4.10 compares the computation speed and the lower bound of two branching rules, and Table 4.11 compares the number of columns and nodes generated by each branching rule and the maximum depth of the branching trees. The designation "$\#inst$." represents the number of instances in the problem class for which the CG algorithm provided fractional solutions and in which the branching was executed. The designations "$\#solv$." and "$\#feas$." denote the number of instances for which each strategy found the optimal solution and the feasible solution, respectively. As mentioned above, when branching over the original variable, the branching divides over an allocation between a UEMS facility and a demand point, which provides an unbalanced tree. This impedes the improvement of the lower bound in the best-first search and the discovery of the feasible integer solution, which can be observed in the table below. When branching over the original variable, even in the smallest problem class, C11, the optimal

solution was found in only two instances among four. Especially in problem C113, more than 150 nodes were generated in 3,600 seconds but could not even find a feasible solution. On the contrary, by applying the Ryan-Foster branching rule, all four instances were solved within the average computation time of 1.04 seconds.

In the small-sized problems, the Ryan-Foster branching rule could solve eight out of 18 instances within 2 seconds, and could solve 16 instances within 3,600 seconds. In the node point-of-view, six and 11 instances were solved using nodes below 10 and 100, respectively. The average of the maximum depth of the Ryan-Foster branching tree was 8.6. By applying the rule of the branching over the original variable to the small-sized problems, four and six instances were solved within 2 and 3,600 seconds, respectively. The remaining 12 instances were not solved within the time limitation. In the node perspective, six out of 18 instances were solved within 100 nodes, and the average of the maximum depth of the branching trees was 15.7. Among 18 instances in the small-sized problems, the branching over the original variables could find the optimal solution faster than the Ryan-Foster branching rule in only two instances, and the Ryan-Foster branching could solve these instances in less than 2 seconds. In the large-sized problems, we tested 10 instances of the problem class C210. Both branching rules could not solve the problems to the optimal. In the lower bound perspective, the branching over the original variables achieved slightly superior results in three out of 10 instances, and the Ryan-Foster branching rule achieved better results in seven instances.

There was no conclusive distinction observed in the size of the nodes. However, for the instances solved to the optimal, the branching over the original variable required the same or bigger number of nodes. It is carefully speculated that the

branching over the original variable would have a larger branching trees when solved to the optimal. Also, the branching over the original variables was relevant to the deeper maximum depth of the branching trees, induced by the unbalanced branching. Although the subproblem of the Ryan-Foster branching is more complicated because of the disjunctive constraints, it had an advantage in branching efficiency. Thus, the Ryan-Foster branching is more suited in the ULAP.

Table 4.10: Computation speeds and lower bounds of the branching strategies.

Problem Class	# *inst.*	# (*solv.*, *feas.*)		*Time*		*Lower bounds*	
		RF.	Org.	RF.	Org.	RF.	Org.
C11	4	(4, 0)	(2, 1)	1.04	1,800.54	1,028,045	925,282*
C13	4	(4, 0)	(1, 3)	20.52	1,882.31	3,151,723	3,097,290*
C23	10	(8, 2)	(2, 1)	974.36*	2,882.20*	3,184,722*	3,091,663*
C210	10	(0, 0)	(0, 0)	3,600*	3,600*	6,269,589*	6,269,521*

$*$: There are instances not solved within the time limit.

Table 4.11: Search tree of the branching strategies.

Problem Class	# *inst.*	*Cols*		*Nodes*		*Max depth*	
		RF.	Org.	RF.	Org.	RF.	Org.
C11	4	225.0	85.3*	13.3	119.5*	5.3	9.3*
C13	4	406.0	162.0*	33.5	106.5*	5.8	21.0*
C23	10	1,089.5*	252.1*	351.8*	132.2*	11.0*	16.2*
C210	10	2,214.2*	1,682.5*	14.0*	15.0*	5.3*	5.0*

$*$: There are instances not solved within the time limit.

In Sections 4.4.3 - 4.4.5, the subproblem of the ULAP and the two solution approaches are proposed. In the computational experiments, a hybrid approach using both MILP reformulation and decomposed DP approaches were used in the B&P algorithm. In the initial stage of the CG process, it was easy to solve the RDCIKP

with the MILP reformulation approach because a large gap existed between the optimal solution of the RMLP and the Lagrangian dual bound. Thus, when the CG was started, the cost-reducing column could be found rapidly with the MILP reformulation approach solved by the commercial solver. The decomposed DP approach took advantage of the special-purpose solver of the knapsack problem. By decomposing the RDCIKP into several nominal 0-1 knapsack problems, we could solve the subproblem with a dynamic programming approach. The decomposed DP approach can provide the solution in a robust manner because the computation time was consistent, regardless of the progress of the CG and the corresponding convergence of the bound.

To benefit from both approaches, a hybrid algorithm was used in this research. In the early stage of the CG process, the MILP reformulation approach was used to solve the pricing subproblem. When the subproblem is solved by the MILP reformulation approach, there was a tendency of the computation time of each iteration to increase in accordance with the CG process. Thus, after each iteration of the CG algorithm, the time was measured. If the time of an iteration exceeded a predefined criterion, the solution algorithm of the subproblem was switched into the decomposed DP approach. The limitation of the time for the change of the solver was chosen arbitrarily to be 100 seconds in the experiment.

In Table 4.12, the average and maximum computation time of iterations in the CG algorithm and the total number of iterations executed within 3,600 seconds are presented. The MILP reformulation approach is named "IP," and the hybrid approach of the "IP" and the decomposed DP approach is named "Mix." We conducted an experiment with 10 instances of problem class C320. During the experiment, the

last iteration started before the time limitation (3,600 seconds) was finished without an interruption. Thus, the maximum computation time can be longer than the time limit.

In Figure 4.3, the progress of the iteration time over the CG process is presented. In the early stage ($\sim$ 5 iterations), the iteration time was plotted on a linear scale, and in the rest of the CG process, a log scale was used. In the figure, the time per iteration of the IP approach increased exponentially as the CG processed. Around the 20th to 25th iteration, instances with the Mix approach switched the solution algorithm from the MILP reformulation approach to the decomposed DP approach. After the conversion, the iteration time became steady, with the similar value of the average computation time of iterations given in Table 4.12. There was a significant difference in the number of the total iterations executed before the time limit between the two approaches, so the corresponding performance of the CG showed an enormous difference. The result showed the necessity of the DP approach solving the subproblem. Also, the hybrid approach was employed in the computational experiments.

Table 4.12: Comparison of the solution approaches of the pricing subproblem.

	Avg. time (*s*)		*Max time* (*s*)		*Iters*	
Problem	IP	Mix	IP	Mix	IP	Mix
C3200	274.42	51.79	3,603.37	116.10	25	70
C3201	161.55	43.07	2,274.74	146.42	26	84
C3202	220.30	43.05	3,900.40	257.56	24	84
C3203	601.87	44.90	12,874.05	392.17	26	81
C3204	151.89	43.72	2,999.24	157.20	25	83
C3205	305.94	48.29	5,838.34	560.62	30	75
C3206	1,062.06	48.62	24,121.49	487.44	24	75
C3207	521.88	48.85	10,285.82	644.72	21	74
C3208	166.00	47.27	1,773.10	303.17	22	77
C3209	240.01	43.57	4,356.96	206.74	28	83

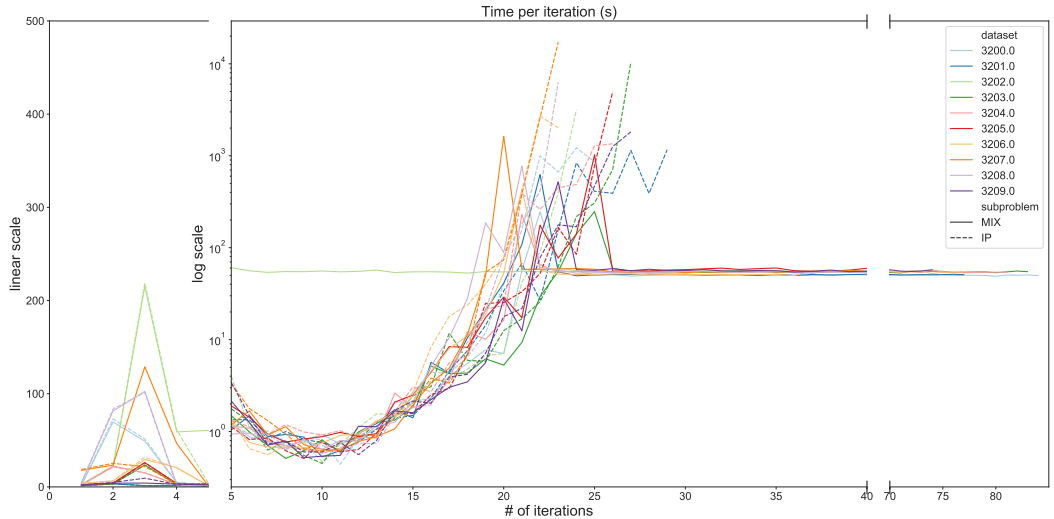


Figure 4.3: Time per iteration of the CG.

4.6 Summary

The advancement of UAV technology is accelerating the introduction of UAVs into various elements of the social system. This paper introduced the location and allocation problem of UAVs in the emergency medical service system considering demand uncertainties. A UAV-operated system would seemingly lead to new operation problems, which would make applying the solution approach from existing literature difficult. The UEMS system, instead of using one or two UAVs, could use a fleet of UAVs operating from a facility. Furthermore, the number of UAVs employed plays an essential role in defining the capability of a facility. The resource availability is modeled to increase gradually along with the number of UAVs, which is considered when allocating demand points to a facility. In the ULAP, both the variable coverage distance and the capacity of the UEMS facility are modeled as functions of the number of UAVs. First, the quadratic constraint of the variable coverage distance is equivalently linearized. The demand is modeled based on the cardinality-constrained uncertainty set, and the resultant nonlinear capacity constraint of the model is reformulated to the MILP model.

However, because of the highly fractional solution of the LP relaxation, the reformulated standard formulation of the ULAP has a weak LP relaxation bound and is challenging to solve with a commercial solver. An extended formulation and a B&P algorithm was proposed in this paper to improve the LP bound and utilize it. Two branching strategies were compared, and the Ryan-Foster branching strategy showed a better performance than the branching over the original variables. This was because of more balanced branching trees achieved through the application of the Ryan-Foster branching strategy. However, the Ryan-Foster branching strategy requires a

disjunctive constraint in the subproblem of the B&P algorithm. The subproblem is defined as a robust disjunctively constrained integer knapsack problem. Based on the existing knowledge about various types of knapsack problems, a decomposed DP approach was proposed as a special-purpose solver. A hybrid approach changing the solution algorithm of the subproblem between the MILP formulation and the decomposed DP approach was also introduced. A restricted master heuristic was proposed for a primal solution based on the EF.

In the computational experiment comparing the performance of the RSF and the EF and the corresponding B&P algorithm, the EF showed stronger LP bounds. Despite the weak LP bound, in the small-sized problems, the RSF could find the optimal solution in a short computation time by virtue of root cutting and heuristics. In the larger problems, the RSF reported a large Gap_L, which represented a slow convergence of the algorithm. Even though the EF showed a better bound, the B&P algorithm could not provide feasible solutions for large-sized problems because the algorithm depended on the best-first search. The restricted master heuristic could provide time-efficient feasible solutions. Mostly the RMH could be initiated regardless of the progress of the B&P algorithm. It was observed that the RMH that started 1,200 seconds after the beginning of the B&P algorithm could provide an efficient solution within a short computation time. In the largest problem class, C320, the RMH could find an even better feasible solution than RSF.

In this research, the RMH was initiated after the discontinuation of the B&P algorithm to measure the sole performance of the RMH. However, it was expected that the RMH could provide the primal bounds if implemented in the middle of the B&P algorithm, and could thereby help the searching process. Also, because

the heuristics in the commercial solver provided effective feasible solutions for RSF, they could be used for the initial columns. Furthermore, other heuristics related to the robust knapsack and the facility location problem with variable coverage could be useful.

Chapter 5

Conclusions and future research

5.1 Summary

The UAV is becoming an essential component of future logistics, which can be operated as a link connecting individual components to the central system. It can be operated as a multipurpose agent in various industry fields, owing to its low price, rapid speed, and flexible operation by the aerial operation. Not only in the logistics but also the public service sector, there are attempts to utilize UAVs. In this dissertation, we proposed two operation problems to construct an emergency wireless network in a disaster situation and one location-allocation problem of the UAV EMS facility. Three problems were modeled based on the set covering approach and solved based on the branch-and-price algorithm.

In Chapter 2, to utilize the advantage of the UAV, the location of the UAV was not bounded to the predefined candidate positions. However, the additional decision on the position weakens the continuous relaxation bound of the problem because the continuous relaxation neutralizes the coverage radius constraint. Despite the quadratic coverage constraint and the continuous decision variable, the generic framework of Dantzig-Wolfe decomposition can be used for the USCP. Based on Dantzig-Wolfe decomposition, the extended formulation and the branch-and-price

algorithm is proposed. The reformulation model could consider the coverage constraint implicitly in the variables, so had improved lower bounds. Based on Jung's theorem, the sufficient condition of a pair of demand points which could be covered by the same UAV is provided. The approximation model could avoid the numerical instability caused by the quadratic coverage constraint. The approximation ratio of the approximation model is provided, along with an approximation model based on the discretization of every lattice point over the plane. The algorithmic performances of the proposed algorithms are compared, and the branch-and-price and its approximation algorithm could solve the realistic-scaled problem within reasonable computation time.

Chapter 3 considered an extra decision on the radius in the coverage constraint. Even though the literature [14] failed to model the problem in explicit mathematical formulation and claimed that a brute force search is required, an explicit mathematical model is proposed with the analysis of its feasible region. Because of the nonconvex feasible set of the continuous relaxation, the extended formulation was not able to be solved by itself. A solvable equivalent problem in mixed-integer linear programming is proposed based on the minimum covering circle, which allowed the implementation of the B&P algorithm. To accelerate the computation speed, a heuristic based on the USCP and a hybrid approach of the heuristic and the exact branch-and-price were provided. Unlike to Berman et al. [14], it is shown that at most $(|N|^3)$ iterations of column generation is required to define every minimal subset, and the experiment showed that much fewer iterations were required to solve the problem. In the computational experiment, the proposed algorithms outperformed the genetic algorithm proposed by Berman et al. [14].

In Chapter 4, the coverage distance of a UAV EMS facility is modeled as a hard constraint to reflect the flight distance constraint of UAVs. The coverage distance is modeled as a function of the number of UAVs assigned to the facility, with a modification of the concept of busy-fraction. The variable coverage distance is set to be the radius of the area that the assigned number of the UAVs can provide a certain level of resource availability, and with the assumption of the demand evenly distribution of the plain, the variable coverage distance was modeled to have a quadratic relation to the number of UAVs. The robust optimization approach of the cardinality-constrained uncertain demand and the capacity related to the number of UAVs resulted in the weak LP bound. The extended formulation and the corresponding branch-and-price algorithm is designed, with a comparison of the various branching strategies. The pricing subproblem was modeled as a robust disjunctively constrained integer knapsack problem, and two solution approaches for the subproblem, mixed-integer linear programming reformulation and the decomposed dynamic programming approaches, were proposed and showed better LP relaxation bounds. For the time-efficient solution of the large-sized problems, a heuristic based on the extended formulation is proposed.

The results of the research can be implemented not only in the UAV application but also in the abstracted problems of clustering and categorization. If the information of the multidimensional data is given and the distance between the data points is defined, the USCP can be used to cluster the data into the minimum number of subsets, where each subset is bounded within a given diameter. In a similar way, the UVCP can categorize the data by minimizing the cost of total subsets. Various possible applications that can utilize the solution approaches proposed in this

dissertation are also suggested for future research.

5.2 Future research

There are several future research on the UAV set covering problem and UAV variable radius set covering problem in Chapters 2 and 3. 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. For the UVCP, the realistic constraints on the flight altitudes can define the minimum and maximum coverage radius. The effects of the size limitations on the coverage radius should be investigated when implementing the system in real applications because the computation speed can drastically change when the coverage distance is limited. Although the battery constraint was not considered in this research, and the problems were approached with the set covering models, it is still one of the essential characteristics of the UAV operation. The routing model, cost structure considering the detailed battery consumption, and scheduling model considering the recharges can be more realistic approaches for the problem. Recent researches are considering location-routing models and inventory-routing models, which can be the possible future researches. In Chapter 3, it is observed that the root node column generation provided the integer solution for every instance we experimented. The solution structure and the progress of the column generation should be investigated for insightful research.

In Chapter 4, more realistic problems could be solved with actual datasets. For an actual cost-benefit analysis, complex cases could be modeled with triage [98],

patient behaviors [59], and multiple levels of hospitals [98]. One could draw input from the existing literature by using more advanced busy-fraction models while defining the function of the variable coverage distance. In this research, a commercial solver was used to solve the subproblem with the MILP reformulation approach without using the structural knowledge. Polytope-associated approaches and cutting plane algorithms could be utilized while solving the RDCIKP. One can refer to Ben Salem et al. [11], who studied the polytope and the facet defining inequalities of the disjunctive-constrained knapsack problem. Atamtürk [6] studied the inequalities on covers and packs of the integer knapsack sets, and Atamtürk [7] focused on the convex hull of the robust knapsack problem. Another research extension related to the healthcare facility location problem is the data-driven and dynamic relocation of UAVs among the UEMS facilities. Cooperation of the UEMS with the existing EMS system would require a detailed analysis before implementing the UEMS in a real-world environment.

Appendices

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

Figures A.1, A.2, A.3, and A.4 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.

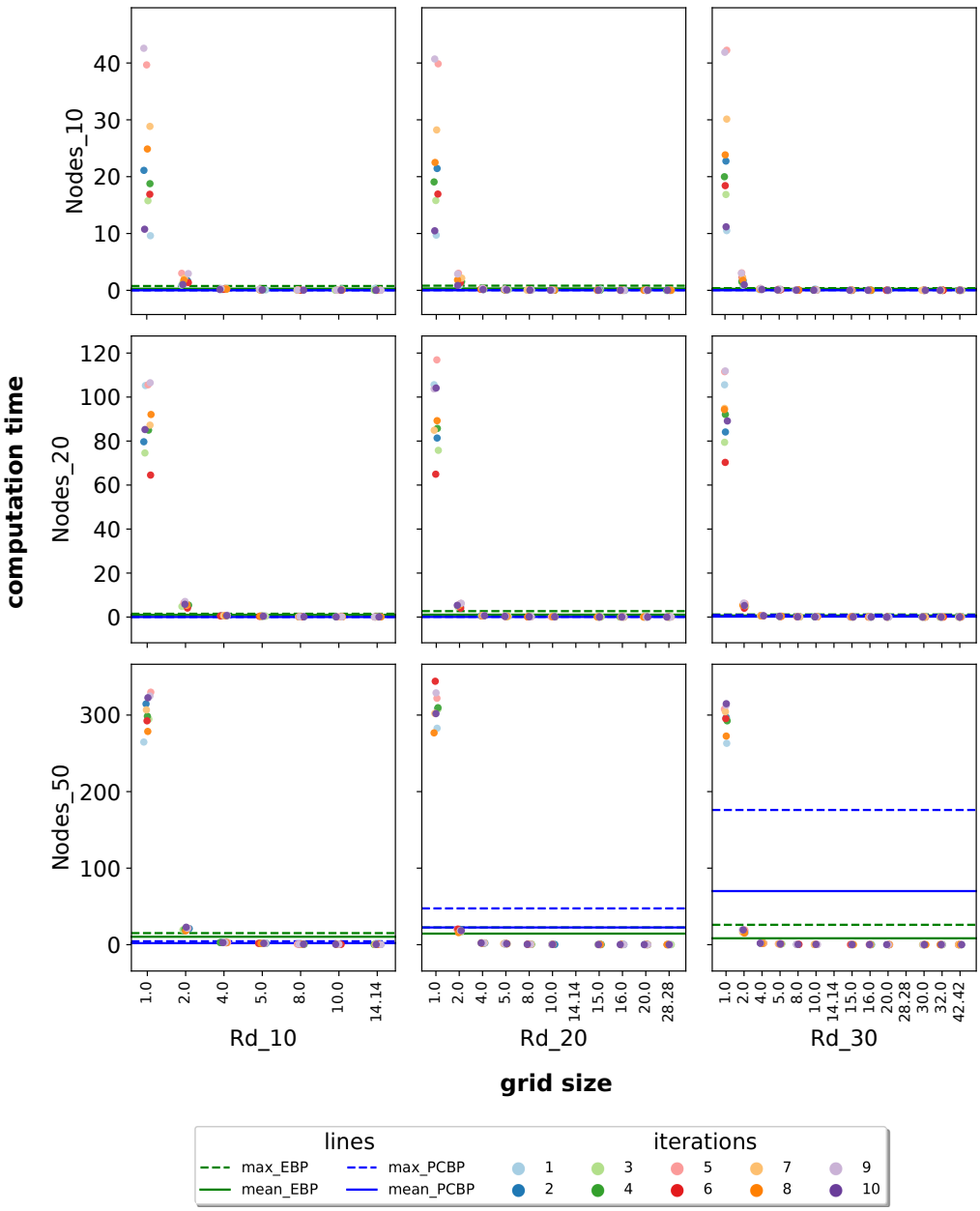


Figure A.1: Computation time of small-sized artificial problems

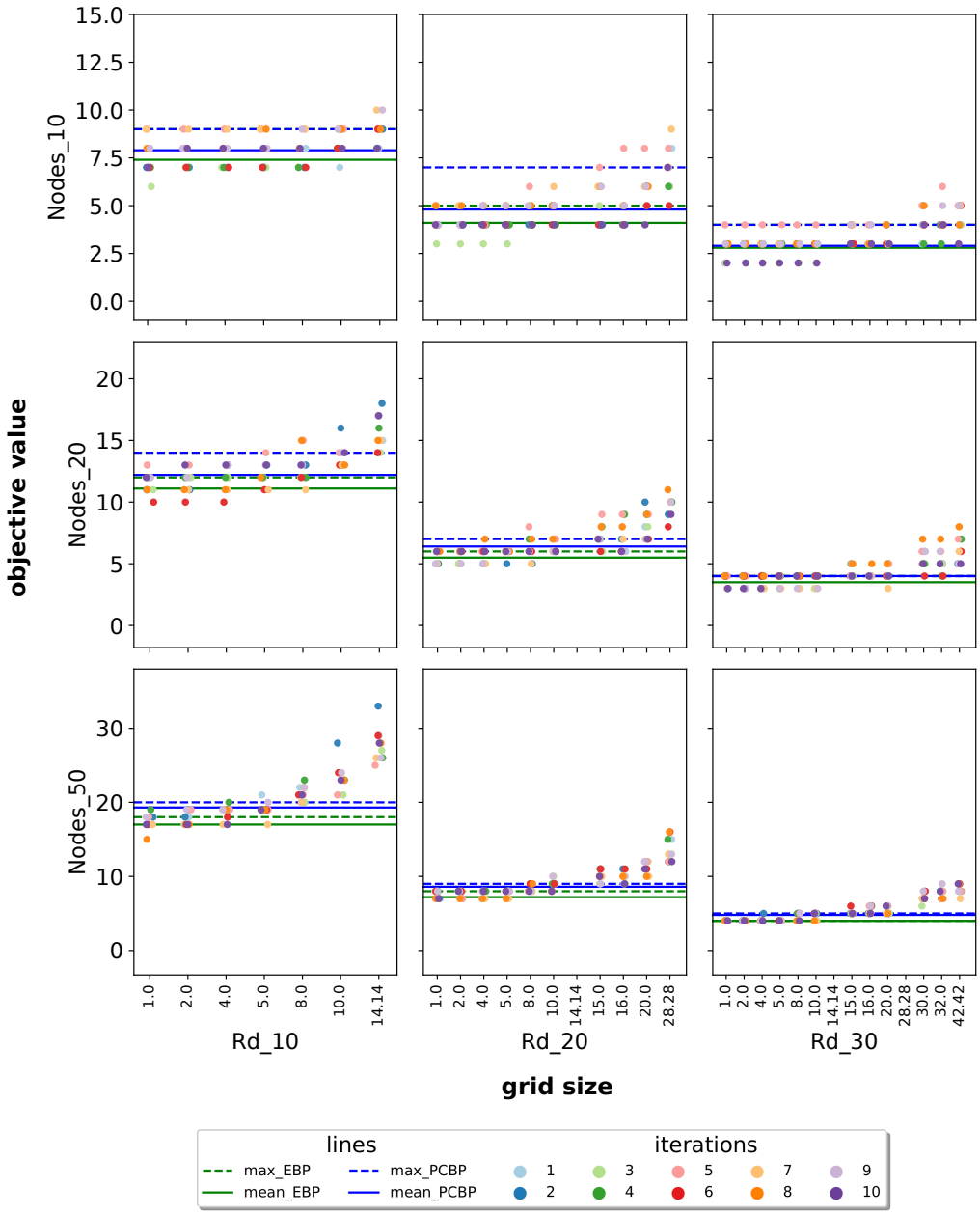


Figure A.2: Objective value of small-sized artificial problems

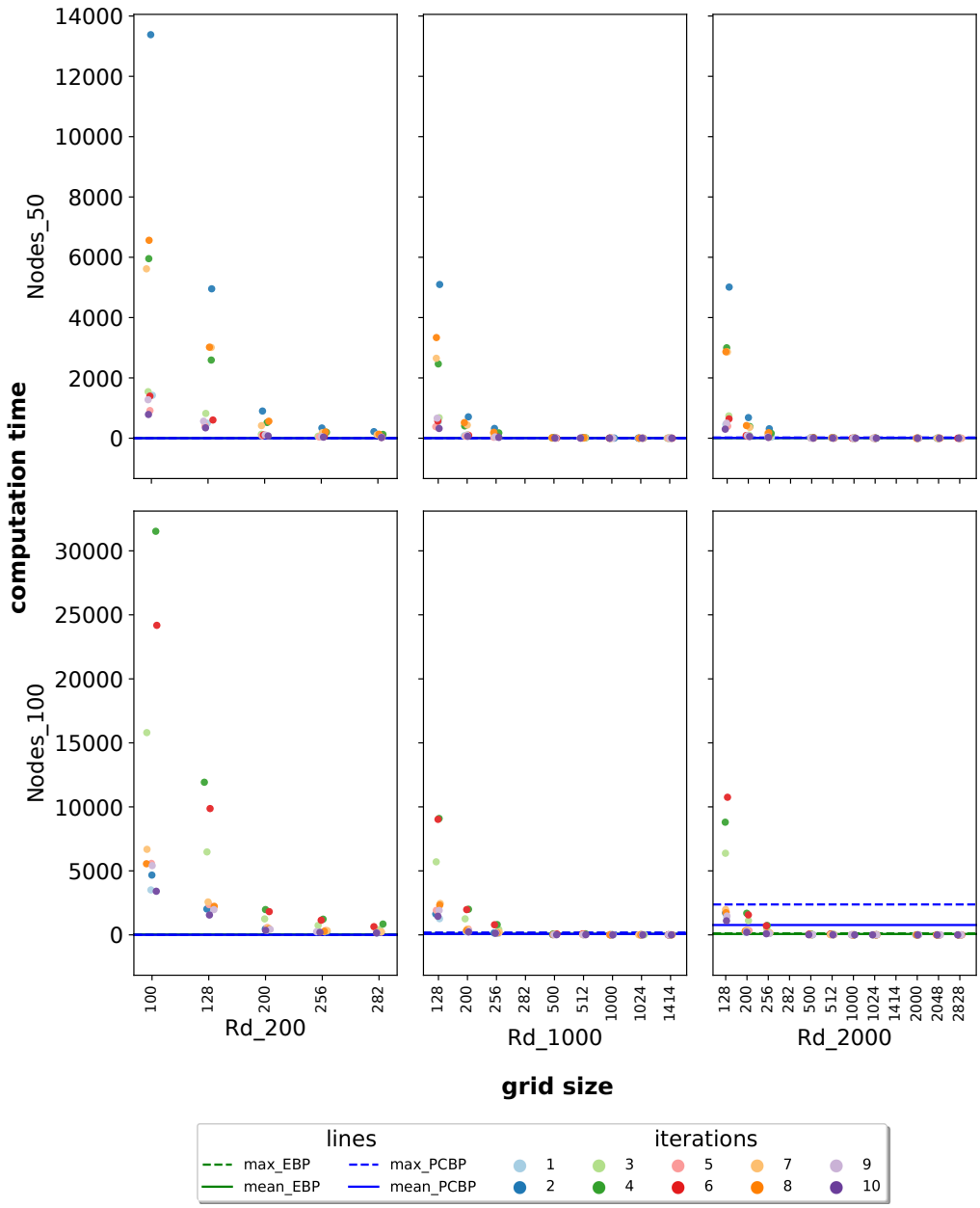


Figure A.3: Computation time of realistic-scale problem

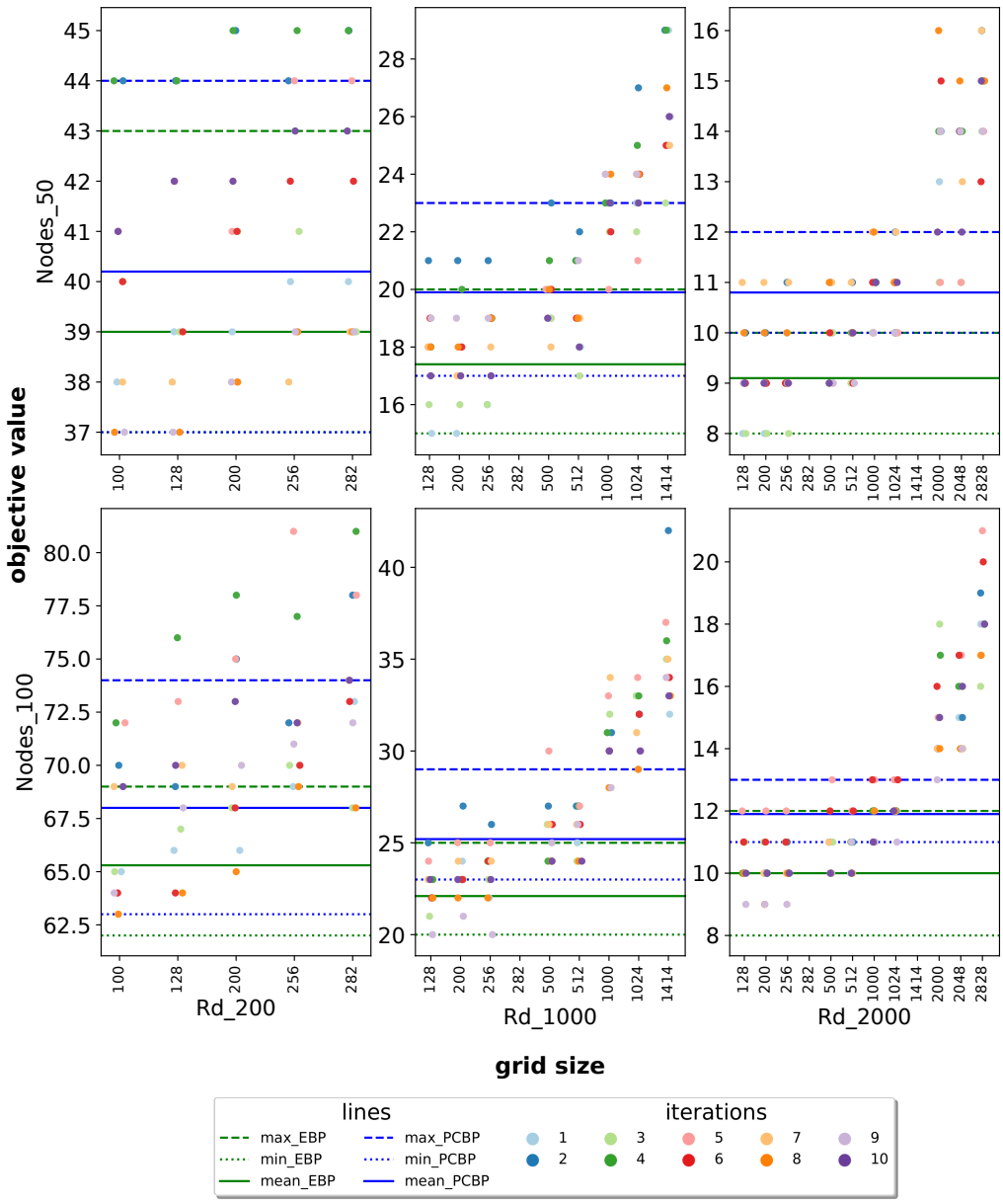


Figure A.4: Objective value of realistic-scale problem

Bibliography

[1] A. Ahmadi-Javid, P. Seyedi, and S. S. Syam, *A survey of healthcare facility location*, Computers & Operations Research, 79 (2017), pp. 223–263.

[2] S. Aida, Y. Shindo, and M. Utiyama, *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, 2013, pp. 19–25.

[3] S. Akl, R. Benkoczi, D. R. Gaur, H. Hassanein, S. Hossain, and M. Thom, *On a class of covering problems with variable capacities in wireless networks*, Theoretical Computer Science, 575 (2015), pp. 42–55.

[4] E. Alekseeva, N. Kochetova, Y. Kochetov, and A. Plyasunov, *Benchmark library "discrete location problems"*, n.d. (accessed October 4, 2020).

[5] R. Aringhieri, M. Bruni, S. Khodaparasti, and J. van Essen, *Emergency medical services and beyond: Addressing new challenges through a wide literature review*, Computers & Operations Research, 78 (2017), pp. 349–368.

[6] A. Atamtürk, *Cover and pack inequalities for (mixed) integer programming*, Annals of Operations Research, 139 (2005), pp. 21–38.

[7] ——, *Strong Formulations of Robust Mixed 0–1 Programming*, Mathematical Programming, 108 (2006), pp. 235–250.

[8] C. Barnhart, E. L. Johnson, G. L. Nemhauser, M. W. P. Savelsbergh, and P. H. Vance, *Branch-and-Price: Column Generation for Solving Huge Integer Programs*, Operations Research, 46 (1998), pp. 316–329.

[9] J. E. Beasley, *OR-Library: Distributing Test Problems by Electronic Mail*, Journal of the Operational Research Society, 41 (1990), pp. 1069–1072.

[10] V. Bélanger, A. Ruiz, and P. Soriano, *Recent optimization models and trends in location, relocation, and dispatching of emergency medical vehicles*, European Journal of Operational Research, 272 (2019), pp. 1–23.

[11] M. Ben Salem, R. Taktak, A. R. Mahjoub, and H. Ben-Abdallah, *Optimization algorithms for the disjunctively constrained knapsack problem*, Soft Computing, 22 (2018), pp. 2025–2043.

[12] P. Beraldi and M. E. Bruni, *A probabilistic model applied to emergency service vehicle location*, European Journal of Operational Research, 196 (2009), pp. 323–331.

[13] O. Berman, Z. Drezner, and D. Krass, *Generalized coverage: New developments in covering location models*, Computers & Operations Research, 37 (2010), pp. 1675–1687.

[14] O. Berman, Z. Drezner, D. Krass, and G. O. Wesolowsky, *The variable radius covering problem*, European Journal of Operational Research, 196 (2009), pp. 516–525.

[15] O. Berman, D. Krass, and Z. Drezner, *The gradual covering decay location problem on a network*, European Journal of Operational Research, 151 (2003), pp. 474–480.

[16] D. Bertsimas and Y. Ng, *Robust and stochastic formulations for ambulance deployment and dispatch*, European Journal of Operational Research, 279 (2019), pp. 557–571.

[17] D. Bertsimas and M. Sim, *Robust discrete optimization and network flows*, Mathematical Programming, 98 (2003), pp. 49–71.

[18] ——, *The Price of Robustness*, Operations Research, 52 (2004), pp. 35–53.

[19] C. Boonmee, M. Arimura, and T. Asada, *Facility location optimization model for emergency humanitarian logistics*, International Journal of Disaster Risk Reduction, 24 (2017), pp. 485–498.

[20] R. Borndörfer and R. Weismantel, *Set packing relaxations of some integer programs*, Mathematical Programming, 88 (2000), pp. 425–450.

[21] L. Brotcorne, G. Laporte, and F. Semet, *Ambulance location and relocation models*, European Journal of Operational Research, 147 (2003), pp. 451–463.

[22] J. Brunkard, G. Namulanda, and R. Ratard, *Hurricane Katrina Deaths, Louisiana, 2005*, Disaster Medicine and Public Health Preparedness, 2 (2008), pp. 215–223.

[23] H. Calik, M. Labbé, and H. Yaman, *p-Center Problems*, in Location Science, G. Laporte, S. Nickel, and F. Saldanha da Gama, eds., Springer, 2015, pp. 79–92.

[24] V. Capoyleas, G. Rote, and G. Woeginger, *Geometric Clusterings*, Journal of Algorithms, 12 (1991), pp. 341–356.

[25] S. Ceria, C. Cordier, H. Marchand, and L. A. Wolsey, *Cutting planes for integer programs with general integer variables*, Mathematical Programming, 81 (1998), pp. 201–214.

[26] G. S. L. K. Chand, M. Lee, and S. Y. Shin, *Drone Based Wireless Mesh Network for Disaster/Military Environment*, Journal of Computer and Communications, 6 (2018), pp. 44–52.

[27] K. Chandrashekar, M. R. Dekhordi, and J. S. Baras, *Providing Full Connectivity in Large Ad-Hoc Networks by Dynamic Placement of Aerial Platforms*, in IEEE MILCOM 2004. Military Communications Conference, 2004., vol. 3, IEEE, 2004, pp. 1429–1436.

[28] I.-C. Choi and S. S. Chaudhry, *The p-median problem with maximum distance constraints: a direct approach.*, Location Science, 1 (1993), pp. 235–243.

[29] A. Claesson, A. Bäckman, M. Ringh, L. Svensson, P. Nordberg, T. Djärv, and J. Hollenberg, *Time to Delivery of an Automated External Defibrillator Using a Drone for Simulated Out-of-Hospital Cardiac Arrests vs Emergency Medical Services*, JAMA, 317 (2017), pp. 2332–2334.

[30] W. J. Comley, *The location of ambivalent facilities: Use of a quadratic zero-one programming algorithm*, Applied mathematical modelling, 19 (1995), pp. 26–29.

[31] G. B. Dantzig and P. Wolfe, *Decomposition principle for linear programs*, Operations Research, 8 (1960), pp. 101–111.

[32] M. S. Daskin, *A Maximum Expected Covering Location Model: Formulation, Properties and Heuristic Solution*, Transportation Science, 17 (1983), pp. 48–70.

[33] ——, *What you should know about location modeling*, Naval Research Logistics, 55 (2008), pp. 283–294.

[34] M. S. Daskin and K. L. Maass, *The p-Median Problem*, in Location Science, G. Laporte, S. Nickel, and F. Saldanha da Gama, eds., Springer, 2015, pp. 21–45.

[35] G. Desaulniers, J. Desrosiers, and M. M. Solomon, *Column Generation*, Springer, 2006.

[36] M. Desrochers, J. Desrosiers, and M. Solomon, *A new optimization algorithm for the vehicle routing problem with time windows*, Operations Research, 40 (1992), pp. 342–354.

[37] M. Desrochers and F. Soumis, *A column generation approach to the urban transit crew scheduling problem*, Transportation Science, 23 (1989), pp. 1–13.

[38] L. Di Puglia Pugliese, F. Guerriero, D. Zorbas, and T. Razafindralambo, *Modelling the mobile target covering problem using flying drones*, Optimization Letters, 10 (2016), pp. 1021–1052.

[39] T. Drezner, *Location of multiple retail facilities with limited budget constraints — in continuous space*, Journal of Retailing and Consumer Services, 5 (1998), pp. 173–184.

[40] Z. Drezner, K. Klamroth, A. Schöbel, and G. O. Wesolowsky, *The Weber Problem*, in Facility Location: Applications and Theory, Z. Drezner and H. W. Hamacher, eds., Springer, jan 2001, pp. 1–36.

[41] H. A. Eiselt and V. Marianov, *Gradual location set covering with service quality*, Socio-Economic Planning Sciences, 43 (2009), pp. 121–130.

[42] K. Elbassioni, S. Jelić, and D. Matijević, *The relation of Connected Set Cover and Group Steiner Tree*, Theoretical Computer Science, 438 (2012), pp. 96–101.

[43] S. Elhedhli, L. Li, M. Gzara, and J. Naoum-Sawaya, *A Branch-and-Price Algorithm for the Bin Packing Problem with Conflicts*, INFORMS Journal on Computing, 23 (2011), pp. 404–415.

[44] J. Elzinga, D. Hearn, and W. D. Randolph, *Minimax Multifacility Location with Euclidean Distances*, Transportation Science, 10 (1976), pp. 321–336.

[45] J. Fernández, B. Pelegri´n, F. Plastria, and B. Tóth, *Solving a Huff-like competitive location and design model for profit maximization in the plane*, European Journal of Operational Research, 179 (2007), pp. 1274–1287.

[46] N. Garg, G. Konjevod, and R. Ravi, *A Polylogarithmic Approximation Algorithm for the Group Steiner Tree Problem*, Journal of Algorithms, 37 (2000), pp. 66–84.

[47] M. Gendreau, G. Laporte, and F. Semet, *Solving an ambulance location model by tabu search*, Location Science, 5 (1997), pp. 75–88.

[48] M. Gendreau, D. Manerba, and R. Mansini, *The multi-vehicle traveling purchaser problem with pairwise incompatibility constraints and unitary demands: A branch-and-price approach*, European Journal of Operational Research, 248 (2016), pp. 59–71.

[49] M. Grötschel and Y. Wakabayashi, *A cutting plane algorithm for a clustering problem*, Mathematical Programming, 45 (1989), pp. 59–96.

[50] Y. Gu, M. Zhou, S. Fu, and Y. Wan, *Airborne WiFi networks through directional antennae: An experimental study*, in 2015 IEEE Wireless Communications and Networking Conference (WCNC), IEEE, 2015, pp. 1314–1319.

[51] H. Gupta, Z. Zhou, S. R. Das, and Q. Gu, *Connected sensor cover: self-organization of sensor networks for efficient query execution*, IEEE/ACM Transactions on Networking, 14 (2006), pp. 55–67.

[52] J. Heinzelman and C. Waters, *Crowdsourcing crisis information in disaster-affected Haiti*, tech. rep., United States Institute of Peace, 2010.

[53] D.-T. Ho, E. I. Grøtli, P. B. Sujit, T. A. Johansen, and J. B. Sousa, *Optimization of Wireless Sensor Network and UAV Data Acquisition*, Journal of Intelligent & Robotic Systems, 78 (2015), pp. 159–179.

[54] K. L. Hoffman and M. Padberg, *Solving Airline Crew Scheduling Problems by Branch-and-Cut*, Management Science, 39 (1993), pp. 657–682.

[55] K. Hogan and C. ReVelle, *Concepts and Applications of Backup Coverage*, Management Science, 32 (1986), pp. 1434–1444.

[56] P. Holley, *'Water is swallowing us up': In Houston, desperate flood victims turn to social media for survival*, 2017. (accessed 9 September 2019).

[57] H. Huang, A. V. Savkin, and C. Huang, *Round Trip Routing for Energy-Efficient Drone Delivery Based on a Public Transportation Network*, IEEE Transactions on Transportation Electrification, 6 (2020), pp. 1368–1376.

[58] L. Huang, J. Li, and Q. Shi, *Approximation algorithms for the connected sensor cover problem*, Theoretical Computer Science, 809 (2020), pp. 563–574.

[59] H. Jang, K. Hwang, T. Lee, and T. Lee, *Designing robust rollout plan for better rural perinatal care system in Korea*, European Journal of Operational Research, 274 (2019), pp. 730–742.

[60] H. Jang and T. Lee, *Demand point aggregation method for covering problems with gradual coverage*, Computers & Operations Research, 60 (2015), pp. 1–13.

[61] X. Ji and J. E. Mitchell, *Finding optimal realignments in sports leagues using a branch-and-cut-and-price approach*, International Journal of Operational Research, 1 (2005), pp. 101–122.

[62] X. Ji and J. E. Mitchell, *Branch-and-price-and-cut on the clique partitioning problem with minimum clique size requirement*, Discrete Optimization, 4 (2007), pp. 87–102.

[63] E. L. Johnson, A. Mehrotra, and G. L. Nemhauser, *Min-cut clustering*, Mathematical Programming, 62 (1993), pp. 133–151.

[64] S. N. Jonkman, B. Maaskant, E. Boyd, and M. L. Levitan, *Loss of Life Caused by the Flooding of New Orleans After Hurricane Katrina: Analysis of the Relationship Between Flood Characteristics and Mortality*, Risk Analysis, 29 (2009), pp. 676–698.

[65] H. Jung, *Ueber die kleinste Kugel, die eine räumliche Figur einschliesst.*, Journal für die reine und angewandte Mathematik, 123 (1901), pp. 241–257.

[66] M. A. Khan, A. Safi, I. M. Qureshi, and I. U. Khan, *Flying ad-hoc networks (fanets): A review of communication architectures, and routing protocols*, in 2017 First International Conference on Latest trends in Electrical Engineering and Computing Technologies (INTELLECT), 2017, pp. 1–9.

[67] D. Kim, K. Lee, and I. Moon, *Stochastic facility location model for drones considering uncertain flight distance*, Annals of Operations Research, 283 (2019), pp. 1283–1302.

[68] S. Kim and I. Moon, *Traveling Salesman Problem With a Drone Station*, IEEE Transactions on Systems, Man, and Cybernetics: Systems, 49 (2019), pp. 42–52.

[69] A. Kumbhar, Guvenç, S. Singh, and A. Tuncer, *Exploiting lte-advanced hetnets and feicic for uav-assisted public safety communications*, IEEE Access, 6 (2018), pp. 783–796.

[70] M. P. Larsen, M. S. Eisenberg, R. O. Cummins, and A. P. Hallstrom, *Predicting survival from out-of-hospital cardiac arrest: a graphic model.*, Annals of emergency medicine, 22 (1993), pp. 1652–1658.

[71] R. C. Larson, *A hypercube queuing model for facility location and redistricting in urban emergency services*, Computers & Operations Research, 1 (1974), pp. 67–95.

[72] C. Lee, K. Lee, K. Park, and S. Park, *Technical Note—Branch-and-Price-and-Cut Approach to the Robust Network Design Problem Without Flow Bifurcations*, Operations Research, 60 (2012), pp. 604–610.

[73] R.-l. Liu, Z.-j. Zhang, Y.-f. Jiao, C.-h. Yang, and W.-j. Zhang, *Study on Flight Performance of Propeller-Driven UAV*, International Journal of Aerospace Engineering, 2019 (2019), p. 6282451.

[74] Y. Liu, Z. Liu, J. Shi, G. Wu, and W. Pedrycz, *Two-Echelon Routing Problem for Parcel Delivery by Cooperated Truck and Drone*, IEEE Transactions on Systems, Man, and Cybernetics: Systems, (2020), pp. 1–16.

[75] B. Maaskant, S. N. Jonkman, and E. Boyd, *Fatalities due to hurricane Katrina (2005)*, tech. rep., TU Delft, 2018.

[76] D. Manerba and R. Mansini, *The Nurse Routing Problem with Workload Constraints and Incompatible Services*, IFAC-PapersOnLine, 49 (2016), pp. 1192–1197.

[77] A. Mehrotra and M. A. Trick, *A Column Generation Approach for Graph Coloring*, INFORMS Journal on Computing, 8 (1996), pp. 344–354.

[78] ——, *Cliques and clustering: A combinatorial approach*, Operations Research Letters, 22 (1998), pp. 1–12.

[79] M. Mozaffari, W. Saad, M. Bennis, and M. Debbah, *Efficient Deployment of Multiple Unmanned Aerial Vehicles for Optimal Wireless Coverage*, IEEE Communications Letters, 20 (2016), pp. 1647–1650.

[80] ——, *Unmanned Aerial Vehicle with Underlaid Device-to-Device Communications: Performance and Tradeoffs*, IEEE Transactions on Wireless Communications, 15 (2016), pp. 3949–3963.

[81] I. H. Osman and N. Christofides, *Capacitated Clustering Problems by Hybrid Simulated Annealing and Tabu Search*, International Transactions in Operational Research, 1 (1994), pp. 317–336.

[82] J. A. Paredes, C. Saito, M. Abarca, and F. Cuellar, *Study of effects of high-altitude environments on multicopter and fixed-wing uavs' energy consumption and flight time*, in 2017 13th IEEE Conference on Automation Science and Engineering (CASE), 2017, pp. 1645–1650.

[83] S. Periyasamy, S. Khara, and S. Thangavelu, *Balanced Cluster Head Selection Based on Modified k-Means in a Distributed Wireless Sensor Network*, International Journal of Distributed Sensor Networks, 12 (2016), pp. 1–11.

[84] U. Pferschy and J. Schauer, *The knapsack problem with conflict graphs.*, Journal of Graph Algorithms and Applications, 13 (2009), pp. 233–249.

[85] F. Plastria, *Continuous Covering Location Problems*, in Facility Location: Applications and Theory, Z. Drezner and H. W. Hamacher, eds., jan 2001, pp. 37–79.

[86] Y. Pochet and L. A. Wolsey, *Integer knapsack and flow covers with divisible coefficients: polyhedra, optimization and separation*, Discrete Applied Mathematics, 59 (1995), pp. 57–74.

[87] C. ReVelle and K. Hogan, *A Reliability-Constrained Siting Model with Local Estimates of Busy Fractions*, Environment and Planning B: Planning and Design, 15 (1988), pp. 143–152.

[88] ——, *The Maximum Availability Location Problem*, Transportation Science, 23 (1989), pp. 192–200.

[89] D. M. Ryan and B. A. Foster, *An Integer Programming Approach to Scheduling*, in Computer Scheduling of Public Transport: Urban Passenger Vehicle and Crew Scheduling, A. Wren, ed., North-Holland, 1981, pp. 269–280.

[90] R. Sadykov and F. Vanderbeck, *Bin Packing with Conflicts: A Generic Branch-and-Price Algorithm*, INFORMS Journal on Computing, 25 (2013), pp. 244–255.

[91] R. Sadykov, F. Vanderbeck, A. Pessoa, I. Tahiri, and E. Uchoa, *Primal Heuristics for Branch and Price: The Assets of Diving Methods*, INFORMS Journal on Computing, 31 (2019), pp. 251–267.

[92] P. Sasikumar and S. Khara, *K-means clustering in wireless sensor networks*, in 2012 Fourth international conference on computational intelligence and communication networks, IEEE, 2012, pp. 140–144.

[93] M. Savelsbergh, *A Branch-and-Price Algorithm for the Generalized Assignment Problem*, Operations Research, 45 (1997), pp. 831–841.

[94] A. V. Savkin and H. Huang, *Range-Based Reactive Deployment of Autonomous Drones for Optimal Coverage in Disaster Areas*, IEEE Transactions on Systems, Man, and Cybernetics: Systems, (2019), pp. 1–5.

[95] S. Sekander, H. Tabassum, and E. Hossain, *Multi-tier drone architecture for 5g/b5g cellular networks: Challenges, trends, and prospects*, IEEE Communications Magazine, 56 (2018), pp. 96–103.

[96] H. Shakhatreh, A. Khreishah, J. Chakareski, H. B. Salameh, and I. Khalil, *On the continuous coverage problem for a swarm of uavs*, in 2016 IEEE 37th Sarnoff Symposium, 2016, pp. 130–135.

[97] S. M. Shavarani, M. G. Nejad, F. Rismanchian, and G. Izbirak, *Application of hierarchical facility location problem for optimization of a drone delivery system: a case study of Amazon prime air in the city of San Francisco*, The International Journal of Advanced Manufacturing Technology, 95 (2018), pp. 3141–3153.

[98] K. Shin and T. Lee, *Emergency medical service resource allocation in a mass casualty incident by integrating patient prioritization and hospital selection problems*, IISE Transactions, 52 (2020), pp. 1141–1155.

[99] T.-P. Shuai and X.-D. Hu, *Connected Set Cover Problem and Its Applications BT - Algorithmic Aspects in Information and Management*, Berlin, Heidelberg, 2006, Springer Berlin Heidelberg, pp. 243–254.

[100] D. Simchi-Levi, X. Chen, and J. Bramel, *The logic of logistics: Theory, Algorithms, and Applications for Logistics and Supply Chain Management*, Springer, 2005.

[101] C. Toregas, R. Swain, C. ReVelle, and L. Bergman, *The Location of Emergency Service Facilities*, Operations Research, 19 (1971), pp. 1363–1373.

[102] P. L. van den Berg, G. J. Kommer, and B. Zuzáková, *Linear formulation for the Maximum Expected Coverage Location Model with fractional coverage*, Operations Research for Health Care, 8 (2016), pp. 33–41.

[103] P. H. Vance, *Branch-and-Price Algorithms for the One-Dimensional Cutting Stock Problem*, Computational Optimization and Applications, 9 (1998), pp. 211–228.

[104] P. H. Vance, C. Barnhart, E. L. Johnson, and G. L. Nemhauser, *Solving Binary Cutting Stock Problems by Column Generation and Branch-and-Bound*, Computational Optimization and Applications, 3 (1994), pp. 111–130.

[105] F. Vanderbeck, *Branching in branch-and-price: a generic scheme*, Mathematical Programming, 130 (2011), pp. 249–294.

[106] F. Vanderbeck and M. W. P. Savelsbergh, *A generic view of Dantzig–Wolfe decomposition in mixed integer programming*, Operations Research Letters, 34 (2006), pp. 296–306.

[107] F. Vanderbeck and L. A. Wolsey, *Reformulation and decomposition of integer programs*, in 50 Years of Integer Programming 1958-2008, M. Jünger, T. M. Liebling, D. Naddef, G. L. Nemhauser, W. R. Pulleyblank, G. Reinelt, G. Rinaldi, and L. A. Wolsey, eds., Springer, 2010, pp. 431–502.

[108] H. Wallop, *Japan earthquake: how Twitter and Facebook helped*, 2011. (accessed 9 September 2019).

[109] C. Wankmüller, C. Truden, C. Korzen, P. Hungerländer, E. Kolesnik, and G. Reiner, *Optimal allocation of defibrillator drones in mountainous regions*, OR Spectrum, 42 (2020), pp. 785–814.

[110] E. Welzl, *Smallest enclosing disks (balls and ellipsoids)*, in New results and new trends in computer science, H. Maurer, ed., Springer, 1991, pp. 359–370.

[111] L. Wu, H. Du, W. Wu, D. Li, J. Lv, and W. Lee, *Approximations for minimum connected sensor cover*, in 2013 Proceedings IEEE INFOCOM, 2013, pp. 1187–1194.

[112] Q. Wu, Y. Zeng, and R. Zhang, *Joint Trajectory and Communication Design for Multi-UAV Enabled Wireless Networks*, IEEE Transactions on Wireless Communications, 17 (2018), pp. 2109–2121.

[113] T. Yamada, S. Kataoka, and K. Watanabe, *Heuristic and exact algorithms for the disjunctively constrained knapsack problem*, Information Processing Society of Japan Journal, 43 (2002), pp. 2864–2870.

[114] Y. Zeng, X. Xu, and R. Zhang, *Trajectory Design for Completion Time Minimization in UAV-Enabled Multicasting*, IEEE Transactions on Wireless Communications, 17 (2018), pp. 2233–2246.

[115] Y. Zeng, R. Zhang, and T. J. Lim, *Throughput Maximization for UAV-Enabled Mobile Relaying Systems*, IEEE Transactions on Communications, 64 (2016), pp. 4983–4996.

[116] C. Zhan, Y. Zeng, and R. Zhang, *Energy-Efficient Data Collection in UAV Enabled Wireless Sensor Network*, IEEE Wireless Communications Letters, 7 (2018), pp. 328–331.

[117] H. Zhang and J. C. Hou, *Maintaining sensing coverage and connectivity in large sensor networks*, Ad Hoc & Sensor Wireless Networks, 1 (2005), pp. 89–124.

[118] Z.-H. Zhang and H. Jiang, *A robust counterpart approach to the bi-objective emergency medical service design problem*, Applied Mathematical Modeling, 38 (2014), pp. 1033–1040.

[119] Z.-H. Zhang and K. Li, *A novel probabilistic formulation for locating and sizing emergency medical service stations*, Annals of Operations Research, 229 (2015), pp. 813–835.

[120] L. Zhen, K. Wang, and H. Liu, *Disaster Relief Facility Network Design in Metropolises*, IEEE Transactions on Systems, Man, and Cybernetics: Systems, 45 (2015), pp. 751–761.

[121] Z. Zhou, S. Das, and H. Gupta, *Connected k-coverage problem in sensor networks*, in Proceedings. 13th International Conference on Computer Communications and Networks (IEEE Cat. No.04EX969), 2004, pp. 373–378.

[122] D. Zorbas, L. Di Puglia Pugliese, T. Razafindralambo, and F. Guerriero, *Optimal drone placement and cost-efficient target coverage*, Journal of Network and Computer Applications, 75 (2016), pp. 16–31.

국문초록

현재, 지역 감시에서 물류까지, 무인항공기의 다양한 산업에의 응용이 주목받고 있다. 특히, 스마트 시티의 개념이 대두된 이후, 무인항공기를 공공 서비스 영역에 활용하여 개별 사회 요소를 연결, 정보와 물자를 교환하고자 하는 시도가 이어지고 있다. 본 논문에서는 공공 서비스 영역에서의 무인항공기 운영 문제를 집합덮개문제 관점에서 모형화하였다. 설비위치결정 및 집합덮개문제 영역에 많은 연구가 진행되어 있으나, 무인항공기를 운영하는 시스템의 경우 무인항공기가 갖는 자유도를 충분히 활용하면서도 무인항공기의 물리적 한계를 고려한 운영 계획을 필요로 한다. 우리는 본 문제와 관련된 기존 연구와 현장이 필요로 하는 기술의 괴리를 인식하였다. 이는 다시 말해, 무인항공기가 가지는 새로운 특성을 고려하면 기존의 문제 해결 방법을 통해 풀기 어렵거나, 혹은 새로운 관점에서의 문제 접근이 필요하다는 것이다.

본 논문에서는 재난이 발생한 지역에 무인항공기를 이용하여 긴급무선네트워크를 구성하는 두가지 문제와, 무인항공기를 이용하여 응급의료서비스를 제공하는 시설의 위치설정 및 할당문제를 제안한다. 확장문제로의 재공식화와 분지평가법을 활용하여, 무인항공기의 활용으로 인해 발생하는 문제 해결 방법의 한계를 극복하고 완화한계를 개선하였다.

공공 서비스 영역에서의 무인항공기 운영, 관련된 기존 연구와 본 논문에서 사용하는 대규모 최적화 기법에 대한 개괄적인 설명, 연구 동기 및 기여와 논문의 구성을 1장에서 소개한다. 2장에서는 무인항공기 집합덮개문제를 정의한다. 무인항공기는 미리 정해진 위치 없이 자유롭게 비행할 수 있기 때문에 더 효율적인 운영이 가능하나, 약한 완화한계를 갖게 된다. Dantzig-Wolfe 분해와 분지평가법을 포함한 대규모 최적화 기법을 통해 완화한계를 개선할 수 있으며, 분지나무의 대칭성을 줄여 실제 규모의 문제를

실용적인 시간 안에 해결할 수 있었다. 수치적 불안정성을 피하기 위하여, 두 가지 선형 근사 모형이 제안되었으며, 이들의 근사 비율을 분석하였다. 3장에서는 무인항공기 집합덮개문제를 일반화하여 무인항공기 가변반경 집합덮개문제를 정의한다. 분지평가법을 적용하면서 해결 가능한 평가 부문제를 제안하였으며, 휴리스틱을 설계하였다. 제안한 풀이 방법들이 기존 연구에서 제안한 벤치마크 유전 알고리즘을 능가하는 결과를 나타내었다. 4장에서는 무인항공기 응급의료서비스를 운영하는 시설의 위치설정 및 할당문제를 정의하였다. 2차 가변반경 범위제약이 선형의 동치인 수식으로 재공식화되었으며, 강건최적화 기법으로 인해 발생하는 비선형 문제를 선형화하였다. 대규모 최적화 기법을 적용하면서, 평가 부문제의 구조를 분석하여 두 가지 풀이 기법과 휴리스틱을 제안하였다.

본 연구의 결과는 무인항공기와 비슷한 특징을 가지는 실제 사례에 적용될 수 있으며, 추상적인 문제로써 다양한 분야에 그대로 활용될 수도 있다.

주요어: 강건최적화, 긴급무선네트워크, 무인항공기, 분지평가법, 설비위치결정문제, 열생성기법, 응급의료서비스, 위치설정 및 할당문제, 집합덮개문제

학번: 2014-21815

감사의 글

아무것도 모르는 미성숙한 어린아이로 대학원에 들어와 7년간 헤아릴 수 없는 분들께 신세를 지고, 한 명의 성인이자 연구자로 졸업을 앞두게 되었습니다. 문일경 교수님께 가장 오랜 시간 지도받은 학생 중 하나로서, 처음 듣고는 깨닫지 못했던 모든 가르침들이 한 해 한 해 쌓일수록 다시금 마음에 와닿는 것을 느끼고 있습니다. 항상 부족했던 제자에게 아낌 없이 베풀어주신 가르침과 지지로, 미욱한 제가 이제 겨우 사회로의 첫 걸음을 내딛을 수 있게 되었습니다. 교수님께서 몸소 실천으로써 가르쳐주신 겸손과 인내, 용기, 주인의식과 책임을 끊임없이 되새기며 사회의 어느 곳에서도 교수님의 자랑스러운 제자가 될 수 있도록 항상 노력하겠습니다.

본 박사학위 논문의 심사를 맡아주신 이덕주 교수님, 박우진 교수님, 최인찬 교수님, 강준규 교수님께 감사의 인사를 드립니다. 이덕주 교수님, 연구가 가지는 의의를 항상 고민하고 읽는 이들에게 명확히 전달할 수 있어야 한다는 가르침을 잊지 않겠습니다. 박우진 교수님, 프로포절에서 주신 조언이 연구의 방향을 결정하는 데 큰 영감이 되었습니다. 앞으로 추상적인 문제와 답만이 아니라 아니라 솔루션을 실제로 적용하기 위한 고민을 놓치지 않겠습니다. 최인찬 교수님, 연구의 중요한 허점을 지적해주셔서 꼭 필요한 내용을 추가할 수 있었습니다. 디테일하게 설명해주신 부분을 고민하면서, 부끄럽고 감사하게도 저의 연구에 대해 더 깊이 이해할 수 있었던 것 같습니다. 강준규 교수님, 공학 분야의 연구는 항상 현장의 문제와 가까이 있어야 한다는 사실을 다시 한 번 깨달을 수 있었습니다. 심사과정에서 주셨던 따뜻한 격려를 품고 자신있게 사회로 나아가겠습니다. 또한 서울대학교에서 가르침을 베풀어주셨던 모든 교수님들께 진심으로 감사드립니다. 학생들에게 지원을 아끼지 않으시고, 부끄러움과 어려움을 깨닫게 하시고, 지식을 넘어 삶을 바라보는 자세를 지도해주신 덕분에 여기까지 성장할 수 있었습니다.

Dear Professor Izabela and Peter, thank you very much for collaborating with me on our research. I owe you a lot for helping and supporting me while visiting back and forth between our countries. Yohanes and Mukund, I will never forget the days and nights you helped me writing my first conference paper. 성인경 박사님, 바쁘신 와중에도, (부끄럽게도 차마 다 반영하지 못한) 중요한 코멘트들을 주셔서 감사합니다. 앞날에 항상 건승하시기를 기원합니다.

연구자의 꿈을 키운 요람이었던 카이스트 산업 및 시스템 공학과의 교수님들께도 감사의 말씀을 올리고 싶습니다. 최병규교수님께서 말씀해주신 "쓰레기통을 뒤지는 연구를 해서는 안 된다"는 것을 가슴에 깊이 새기고, 부끄러움을 느끼지 못하는 연구자가 되는 것을 끊임없이 경계하겠습니다. 언제나 그리운 카이스트의 모든 소중한 인연들께도 감사드립니다.

서로 다른 여러 분야에서 공부하고 일하고 있는 고등학교 친구들과의 이야기는 산업공학이라는 분야를 넘어 넓은 시각으로 세계를 볼 수 있게 해주었습니다. 이런 훌륭한 친구들과 꿈과 우정을 함께 키울 수 있도록 가르쳐주신 고등학교 선생님들, 황석규 선생님께 감사의 말씀을 전하고 싶습니다.

고마운 연구실 친구들, 먼저 졸업한 상윤이, 영철형, 여러분의 부재가 저를 졸업할 수 있도록 끊임없이 채찍질하였습니다. 광형, 오랫동안 저의 핸드폰 배경화면을 장식했던 꾸짖음 잊지 않겠습니다. 형의 행복을 기원합니다. 어쩌다 보니 함께 졸업하게 된 동료들, 마지막 학기 서로 속도 많이 썩였지만 너무 많은 도움 또한 받았습니다. 포기하지 않고 함께 해주어 고맙습니다. 정말 오랜 시간동안 연구실에서 함께 먹고 자고 밤을 샌 친구들, 졸업한 모두들, 서로의 고민을 들어주고 일을 기꺼이 나누고, 모르는 것을 알려주고, 지치고 나태해지면 꾸짖어주며 기쁨과 슬픔을 나누었던 소중한 시간을 평생 잊지 않겠습니다. 자랑스러운 후배들, 제가 연구실의 메인이었던 시절보다 다들 공부도 열심히 하고, 일도 잘 해주어 연구실에 대한 한 점의 걱정 없이 졸업할 수 있습니다. 어려울 수 있는 대학원 생활을 최고의 연구실에서 행복하게 보내게 해주어 고맙습니다.

모두의 앞날에 기쁨과 성공만이 있기를 기원합니다. 서울대에서 만난 모든 분들께서 대가 없는 수 많은 호의와 도움을 주셨습니다. 감사한 마음을 지니고 저도 사회에서 모든 이에게 이 은혜를 갚으며 살아가겠습니다.

마지막으로 30년의 시간동안 흔들림 없이 저를 지지하고 지원해주셨던 사랑하는 가족들에게 감사의 마음을 전합니다. 긴 세월이 지나 이제야 큰아들로, 큰형으로 한 사람의 몫을 할 수 있을 것 같습니다. 부모님의 짐을 나눠 지고, 동생들이 멋지게 나아갈 길을 앞에서 조금이라도 틔울 수 있는 든든한 첫째가 되겠습니다. 감사합니다.

2021년 1월

박영수 올림