IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.DOI

Optimizing the strategic decisions for one-way station-based carsharing systems: A mean-CVaR approach

KAI ZHANG[1], YUICHI TAKANO[2], YUZHU WANG [1], AND AKIKO YOSHISE[2]
[1]Graduate School of Systems and Information Engineering, University of Tsukuba, Tsukuba, Ibaraki 305-8573, Japan
[2]Faculty of Engineering, Information and Systems, University of Tsukuba, Tsukuba, Ibaraki 305-8573, Japan

Corresponding author: Kai Zhang (e-mail: mjm17102@gmail.com).

This work was supported by the joint program of the University of Tsukuba and Toyota Motor Corporation, titled "Research on the next generation social systems and mobilities". We are deeply grateful to Toyota Motor Corporation for providing the data and valuable comments on our work. Thanks are also given to China Scholarship Council (CSC) for providing financial support to the author Kai Zhang during his Ph.D. study.

ABSTRACT The study focuses on the strategic decisions including on the location and capacity of stations and the fleet size for designing the one-way station-based carsharing systems. Under demand uncertainty, we introduce a two-stage risk-averse stochastic model to maximize the mean return and minimize the risk, where the conditional value-at-risk (CVaR) is specified as the risk measure. To solve the problem efficiently, a branch-and-cut algorithm and a scenario decomposition algorithm are developed. We conduct computational experiments based on historical use data and generate efficient frontiers so that the system operator can make a trade-off between return and risk. We then utilize an evaluation method to analyze the necessity of introducing risk. Finally, the efficiency of the proposed algorithms is elaborated through comparative experiments. Both branch-and-cut algorithm and scenario decomposition algorithm can tackle the small- and medium-scale problems well. For large-scale problems that cannot be solved by using an optimization solver or the branch-and-cut algorithm, scenario decomposition method can provide favorable solutions within a reasonable time.

INDEX TERMS Branch-and-cut, carsharing, risk, scenario decomposition, stochastic model, strategic decisions.

I. INTRODUCTION

THE continuous increase in the number of private vehicles in the past few decades has had serious negative impacts, such as environmental pollution, time wastage, and shortage of parking spaces. Although traditional multimodal public transportation can satisfy most of the travel demand, many users prefer private vehicles due to their greater convenience, especially people who resides far away from public transportation stations. However, the cost of owning a car and parking difficulties bring their own challenges. Carsharing is receiving increasing attention around the world as a promising sustainable means of transportation that offers the advantages of both private vehicles and public transportation [1]. Such systems are also regarded as a potential solution to the first- and last-mile problem.

Current carsharing systems can be mainly classified into one-way and two-way types. In one-way systems, users can pick up and return cars at different sites, while in two-way systems, users should return rented cars to the site where they were picked up. Compared with two-way systems, one-way systems are more convenient for users, considering that one-way trips usually occupy a large proportion of the total trips [2], [3]. Furthermore, station-based and free-floating systems can be distinguished in view of parking-spot restrictions. The former type forces people to park vehicles at stations with limited parking spaces, whereas the latter allows the users to park cars anywhere in an operation area. In recent years, station-based carsharing systems equipped with charging piles have become popular with the advent of environmentally friendly electric vehicles (EVs).

Design and management of a carsharing system raises several decision-making problems, ranging from strategic issues (e.g., carsharing mode, location) to operational policies (e.g., vehicle relocation, staff scheduling) [4], [5]. In this

study, we focus on the strategic design of one-way station-based carsharing systems from the vantage point of the system operator. The most related research (e.g., [4], [6], [7]) copes with strategic problems based on the traditional risk-neutral two-stage stochastic programming by considering the expectation value as the preference criterion. However, the resulting decisions may be poor under certain realizations of random data. For non-repetitive decision-making problems, such as location planning and network design, a risk-averse approach would provide more robust solutions [8]. Therefore, we analyze the downside risk, which refers to the financial risk of the actual return being below the expected return, in this study. With the objective of maximizing the return and minimizing the risk, we propose a two-stage risk-averse stochastic mixed integer nonlinear programming (MINLP) model, where the conditional value-at-risk (CVaR) is specified as the risk measure, to optimize strategic decisions involving station locations, station capacities, and fleet sizes for one-way station-based carsharing systems. With a training and testing method, we also show the advantage of the risk-averse model under stochastic demand. In order to solve the two-stage stochastic MINLP model, two customized methods are developed: a branch-and-cut algorithm and a scenario decomposition algorithm. The model and algorithms are verified on scenario demand data generated from historical data of Ha:mo RIDE Toyota in Japan.

The remainder of the paper is as follows. Section II presents an overview of previous related studies on carsharing systems. Section III provides an elaborate model formulation for making strategic decisions. Section IV is devoted to the two algorithms for solving the problem. Subsequently, we report computational results in Section V and conclude this study in Section VI.

II. LITERATURE REVIEW

The problems affecting carsharing can be divided into strategic, tactical, and operational dimensions. Generally, the overall structure of a carsharing system is determined at the strategic and tactical levels, which have a significant impact on system performance. Operational problems are usually handled to improve system management. Regarding optimization problems at different levels, interested readers can refer to [9] and [10] for a comprehensive review. More recently, Illgen and Höck [5] presented a systematic review centered around a key operational issue: the vehicle relocation problem (VReP) in one-way carsharing systems and introduced dependent-decision problems at other levels associated with VReP. Over the past decade, VReP has become the most commonly considered problem [11] – [18]. However, only a little literature has concentrated on strategic decisions involving location problems, as revealed by Çalık and Fortz [7].

The classic facility location problems (e.g., covering, median, center problems) have been researched for more than half a century, and there are a large number of related studies. Daskin [19] provided a thorough review in terms of models, algorithms, and applications. Within the context of carsharing, de Almeida Correia and Antunes [20] are pioneers who first solved the station location problem in one-way systems. In their study, three mixed integer linear programming (MILP) models were proposed and compared to determine the optimal number, location, and size of stations with the same objective that maximizes the profit. By modifying the constraints, they incorporated different schemes (i.e., serve some selected requests, serve all requests, and reject some requests conditionally) into the models. The practicality of the models was illustrated with a case study on Lisbon, Portugal. Following this study, Jorge *et al.* [21] developed an agent-based simulation model considering demand variability and vehicle relocation policy. Experiments were conducted on the Lisbon data to test the validity of the solution from the MILP model proposed by de Almeida Correia and Antunes [20]. Similarly, de Almeida Correia *et al.* [3] evolved the MILP formulation of de Almeida Correia and Antunes [20] to allow users to pick up the sharing vehicles at an alternate station other than the closest one. Later, Boyacı *et al.* [4] formulated a multi-objective MILP model for planning one-way electric carsharing systems considering vehicle relocation and EV charging requirements. The objectives of the model are to maximize the profit of the operator and the profit of the users. Because of the large number of relocation variables, the problem becomes intractable in real-world cases. The relocation variables were therefore reduced by grouping the demand and clustering the stations to form an aggregate model. The proposed approach was validated with data from Nice, France. Huang *et al.* [22] showed an MINLP model to fathom the station location and capacity problem with a nonlinear demand rate represented by the logit model. To solve the MINLP model, a customized gradient algorithm was proposed. The model and algorithm were applied to Suzhou Industrial Park, Suzhou, China.

Additionally, some recent studies began to take the demand uncertainty into account when making strategic decisions, whereas the most of the earlier carsharing research including demand uncertainty focused on dealing with VReP. Brandstätter *et al.* [6] identified the optimal locations and the number of required EVs for one-way station-based carsharing systems by introducing a stochastic optimization model whose objective is to maximize the expected profit. In that model, the uncertain demand is represented by several scenarios. Due to the complexity of the problem, they proposed a heuristic algorithm to obtain an approximate solution or to provide an initial heuristic solution. Lu *et al.* [23] introduced a stochastic model to optimize the profitability and quality of service (QoS) considering uncertain one-way and two-way rental demand for a hybrid reservation-based and free-floating system. A desired QoS level was maintained by minimizing the expected penalty of unserved customers in a risk-neutral model. To solve the stochastic model, they developed branch-and-cut algorithm with mixed-integer rounding-enhanced Benders cuts. They also made a brief remark on a risk-averse model by penalizing the CVaR of unserved

demand but did not give corresponding detailed experimental results. Çalık and Fortz [7] modelled the location problem as a two-stage stochastic MILP model and developed a Benders decomposition algorithm to solve it. As in Brandstätter *et al.* [6], the objective function maximized the expected profit. On the basis of a demand prediction model, many demand scenarios were generated from Manhattan taxi trip data and numerical experiments were conducted. The results illustrated that the proposed algorithm could help reduce computing time and obtain favorable solutions.

In conclusion, we can see that there has been little research on determining simultaneously the location and capacity of stations and the fleet size, while considering stochastic demand. Some studies like [4], [6], [7] merely took part of them into account and ingored the potential risk. Although Lu *et al.* [23] briefly introduced a model that applied CVaR to reduce the risk of unserved demand, rare carsharing studies have considered the risk to avoid heavy losses. Thus, to fill in the research gaps, this paper considers a new risk-averse stochastic model to solve the joint design problem. Moreover, with respect to the solution methods, Lu *et al.* [23] solved a similar problem by an algorithm based on Benders decomposition. Our study, however, develops two other algorithms to handle the proposed model and compares the computation efficiency with the algorithm illustrated by Lu *et al.* [23].

III. STRATEGIC DECISION MODEL

A. ASSUMPTIONS

In this study, we determine the station locations, station capacities, and fleet sizes to design one-way station-based carsharing system with a risk-averse model. To formulate the model, some underlying assumptions are as follows.

- The trip demand data are available or predictable in advance, including for different scenarios. Each scenario represents one possible day with a set of estimated trips. The probability is the same for each scenario. Each trip is a tuple of four elements: origin, destination, departure time, and arrival time. The price of each trip can be computed on the basis of the information above.
- For the trip demand to be satisfied, there must be vehicles available at the origin and parking spaces at the destination.
- The working time of the carsharing system is divided into equal time intervals (5 minutes), so we can gather together the trips with the same departure or arrival time interval. In this way, the problem size can be reduced.
- All stations have at least one parking space and have individual maximal capacities that depend on the local conditions. The station cost consists of land, construction, and charging-pile costs. The unit land costs vary depending on the location, but the unit construction and charging-pile costs are fixed.
- Considering the station and vehicle costs tend to be too high to get a profitable system, the profit return in our model only depends on the trip revenue and operating cost. Alternatively, the station and vehicle costs are restricted by a budget constraint. Çalık and Fortz [7] addressed a similar issue by introducing a cost factor to forcibly reduce the cost. This implies the operator must study the pricing policy and other income resources comprehensively, but this issue is out of the scope of this paper.
- The state of charge (SoC) of the EV's battery is ignored in this study. In the case of Ha:mo RIDE Toyota, the vehicles are mostly used for short trips, and a previous study [24] showed that no vehicle ever became unavailable in the system because of a low SoC. Therefore, the proposed model is more suitable for short-range carsharing systems to counter the last-mile problem.
- The system operator does not take into account operational activities (e.g., relocation, staff allocation, or staff scheduling) when making strategic decisions.

B. NOTATION

Before providing the detailed model formulation, we should explain the notation used in the model.

1) Sets and indices

- $s \in S$: scenarios
- $i \in I_s$: trips in scenario s
- $t \in T$: time intervals
- $j \in J$: potential location sites

2) Parameters

- $start_i, end_i \in T$: start- and end-intervals of trip i
- $origin_i, dest_i \in J$: origin and destination of trip i
- P_i^s: charging price for trip i in scenario s
- C_1: operating cost per parking space per scenario
- C_2: operating cost per vehicle per scenario
- H_j: cost for setting a parking space at site j, including land, construction, and charging-pile costs
- F: cost of purchasing a vehicle
- B: available budget
- M_j: maximum number of parking spaces that can be set at site j
- λ: weight value, ranging from 0 to 1
- β: confidence level

3) Variables

- $\alpha \in \mathbb{R}$: auxiliary variable for obtaining minimum β-CVaR
- $p_j \in \mathbb{Z}_+$: number of parking spaces at site j
- $v \in \mathbb{Z}_+$: number of vehicles in the system
- $n_{j,t}^s \in \mathbb{Z}_+$: number of vehicles at site j at the beginning of time interval t in scenario s
- $z_i^s \in \{0, 1\}$: binary variable, if trip i in scenario s is served, the value is 1; otherwise 0

C. MATHEMATICAL FORMULATION

In this part, we present the two-stage risk-averse stochastic MINLP model in the deterministic equivalent form. The first-

stage constraints (1b) and (1c) are given by using variables α, v and p_j, that is, first-stage decision variables. The second-stage decision variables z_i^s and $n_{j,t}^s$, associated with scenario s, are restricted by the second-stage constraints (1d)–(1g).

$$\min \frac{\lambda}{|S|} \sum_{s\in S} \left(-\sum_{i\in I_s} P_i^s z_i^s + C_1 \sum_{j\in J} p_j + C_2 v \right) + (1-\lambda) \left(\alpha + \frac{1}{(1-\beta)|S|} \times \sum_{s\in S} \left[-\sum_{i\in I_s} P_i^s z_i^s + C_1 \sum_{j\in J} p_j + C_2 v - \alpha \right]_+ \right) \tag{1a}$$

subject to:

$$\sum_{j\in J} H_j p_j + F v \le B \tag{1b}$$

$$p_j \le M_j, \forall j \in J \tag{1c}$$

$$n_{j,t+1}^s = n_{j,t}^s - \sum_{\substack{i:origin_i=j \\ start_i=t}} z_i^s + \sum_{\substack{i:dest_i=j \\ end_i=t}} z_i^s, \quad \forall s \in S, j \in J, t \in T \setminus \{t_{\text{last}}\} \tag{1d}$$

$$n_{j,t}^s \ge \sum_{\substack{i:origin_i=j \\ start_i=t}} z_i^s, \forall s \in S, j \in J, t \in T \tag{1e}$$

$$p_j - n_{j,t}^s \ge \sum_{\substack{i:dest_i=j \\ end_i=t}} z_i^s, \forall s \in S, j \in J, t \in T \tag{1f}$$

$$\sum_{j\in J} n_{j,1}^s = v, \forall s \in S \tag{1g}$$

$$\alpha \in \mathbb{R} \tag{1h}$$

$$p_j \in \mathbb{Z}_+, \forall j \in J \tag{1i}$$

$$v \in \mathbb{Z}_+ \tag{1j}$$

$$z_i^s \in \{0,1\}, \forall s \in S, i \in I_s \tag{1k}$$

$$n_{j,t}^s \in \mathbb{Z}_+, \forall s \in S, j \in J, t \in T \tag{1l}$$

The objective function (1a) minimizes the weighted sum of the measures of profitability (expected loss, i.e., the opposite of the expected return) and risk (CVaR). Note that $[x]_+ = \max\{x, 0\}$, for $x \in \mathbb{R}$. The loss function in each scenario is equal to the operating costs of the built parking spaces and purchased vehicles ($C_1 \sum_{j\in J} p_j + C_2 v$) reduced by the revenue of served trips ($\sum_{i\in I_s} P_i^s z_i^s$). In Appendix A, we present a detailed explanation of the objective function. Constraint (1b) states the costs for setting up parking spaces and purchasing cars are within the available budget. Constraints (1c) show the maximal capacity restriction at each potential location site. In addition, there are additional constraints depending on the scenario. Constraints (1d) are for vehicle flow conservation at each station for each time interval. The departure time and arrival time of each trip are known parameters in the model, so no additional constraints are required to represent the travel time. Constraints (1e) ensure there are enough vehicles for trips served at each station and each time interval. Constraints (1f) restrict the number of vehicles arriving at a station to the number of available parking spaces at that station during each time interval, and at the same time, ensure there are enough parking spaces for the vehicles at site j. Constraints (1g) indicate that the total number of vehicles in the system is always equal to the number of purchased vehicles. Constraints (1h)–(1l) are variable restrictions.

IV. SOLUTION METHODS

Due to plenty of variables and constraints, the strategic decision problem becomes more difficult when the number of scenarios increases. To deal with the problem efficiently, we present two solution methods including the branch-and-cut algorithm and scenario decomposition algorithm in this section. The former algorithm mainly focuses on the nonlinear CVaR function, while the latter one pays attention to the special block-angular structure of the stochastic problem.

A. BRANCH-AND-CUT ALGORITHM

Takano *et al.* [25] proposed two cutting-plane algorithms to handle the CVaR function in a portfolio optimization problem with a nonconvex transaction cost. Here, we present a similar algorithm based on the same idea, that is, repeatedly solving the relaxed problems and gradually approximating the CVaR function with a portion of cutting-plane representation.

To formulate the algorithm, we equivalently rewrite the primal problem by introducing the auxiliary variable u.

$$\min \frac{\lambda}{|S|} \sum_{s\in S} \left(-\sum_{i\in I_s} P_i^s z_i^s + C_1 \sum_{j\in J} p_j + C_2 v \right) + (1-\lambda)\, u \tag{2a}$$

subject to:

$$u \ge \alpha + \frac{1}{(1-\beta)|S|} \times \sum_{s\in S} \left[-\sum_{i\in I_s} P_i^s z_i^s + C_1 \sum_{j\in J} p_j + C_2 v - \alpha \right]_+ \tag{2b}$$

$$u \in \mathbb{R} \tag{2c}$$

(1b)–(1l)

According to the proof in [26], the CVaR constraint (2b) is equivalent to the following cutting-plane representation.

$$u \ge \alpha + \frac{1}{(1-\beta)|S|} \times \sum_{s\in \mathcal{H}} \left(-\sum_{i\in I_s} P_i^s z_i^s + C_1 \sum_{j\in J} p_j + C_2 v - \alpha \right), \quad \forall \mathcal{H} \subseteq S \tag{3}$$

Representation (3) obviously contains a series of linear constraints, and the number of constraints is the number of subsets of the scenario set S, i.e., $2^{|S|}$. Since many of the constraints may be redundant, we can simply append

the necessary constraints to the relaxed problem iteratively, instead of using all of them directly. The initial relaxed problem of formulation (2) is arrived at by replacing the CVaR constraint (2b) with constraint (4),

$$u \geq U_{min}. \tag{4}$$

where U_{min} is a sufficiently small constant to prevent the problem from being unbounded. Accordingly, the feasible region of the initial relaxed problem can be defined by

$$\Omega := \{(\alpha, v, \boldsymbol{p}, \boldsymbol{z}, \boldsymbol{n}) : \text{(1b)–(1l), (2c), (4)}\}, \tag{5}$$

where $\boldsymbol{p}$, $\boldsymbol{z}$, and $\boldsymbol{n}$ are the sets of variables p_j, z_i^s, and $n_{j,t}^s$, respectively.

Different from the cutting-plane algorithms in [25], we will not pursue an optimal solution at each iteration, because it takes too much time to execute the branch-and-bound algorithm completely when solving the relaxed MILP problem at each iteration. Instead, we add chosen constraints dynamically to the problem during the branch-and-bound procedure, which is known as *lazy constraint callback*. This callback function is available in optimization software (e.g., Gurobi, CPLEX) and may help to reduce the computing time. With the function, we develop a branch-and-cut algorithm here. The algorithm starts by using the branch-and-bound algorithm to solve the relaxed problem. If a better feasible solution $(\widehat{\alpha}, \widehat{v}, \widehat{u}, \widehat{\boldsymbol{p}}, \widehat{\boldsymbol{z}}, \widehat{\boldsymbol{n}})$ in Ω is found, it checks whether the corresponding *MIP Gap* is within tolerance. The *MIP Gap* is a relative gap between the upper and lower objective bounds in the branch-and-bound algorithm, which can be obtained directly with optimization software. As long as the *MIP Gap* is unqualified, the callback procedure will be activated. When the feasible solution violates the CVaR constraint (2b), cutting planes are generated to separate it from the feasible set. The cut is expressed as:

$$\begin{aligned} u \geq \alpha + \frac{1}{(1-\beta)|S|} \times \\ \sum_{s \in \widehat{\mathcal{H}}} \left(-\sum_{i \in I_s} P_i^s z_i^s + C_1 \sum_{j \in J} p_j + C_2 v - \alpha \right), \end{aligned} \tag{6}$$

where $\widehat{\mathcal{H}} := \{s \in S : -\sum_{i \in I_s} P_i^s \widehat{z}_i^s + C_1 \sum_{j \in J} \widehat{p}_j + C_2 \widehat{v} - \alpha > 0\}$.

Algorithm 1 is a description of our method.

We also prove the convergence of the branch-and-cut algorithm.

Theorem 1. *For any $\epsilon \geq 0$, the branch-and-cut algorithm has the finite convergence property.*

Proof. Suppose that **Algorithm 1** does not converge in a finite number of iterations. Then *MIP Gap* $> \epsilon$ holds for any sufficiently large number of iterations k such that $k > 2^{|S|}$. Let us suppose that *MIP Gap* $> \epsilon$ holds at iteration $k > 2^{|S|}$. Since the integer variables are bounded in our model, at **Step 2** of iteration k, we can find either a feasible solution $(\widehat{\alpha}, \widehat{v}, \widehat{u}, \widehat{\boldsymbol{p}}, \widehat{\boldsymbol{z}}, \widehat{\boldsymbol{n}}) \in \Omega$ that violates the CVaR constraint (2b) or an optimal solution of the current model, within a finite number of branch-and-bound iterations. Suppose that we obtain a feasible solution $(\widehat{\alpha}, \widehat{v}, \widehat{u}, \widehat{\boldsymbol{p}}, \widehat{\boldsymbol{z}}, \widehat{\boldsymbol{n}}) \in \Omega$ violating the CVaR constraint at iteration k. Since we have assumed $k > 2^{|S|}$, this solution satisfies all $k - 1 \geq 2^{|S|}$ inequalities generated before iteration k. Note that the same cutting plane will never be appended twice and the number of cutting planes that can be generated is at most $2^{|S|}$. Thus, the current feasible solution should satisfy all possible cutting planes, which contradicts the assumption that the solution violates the CVaR constraint. On the other hand, if we obtain an optimal solution of the current model, optimality implies that the current *MIP Gap* $\leq \epsilon$ at iteration k, which is a contradiction to the assumption that *MIP Gap* $> \epsilon$ holds. □

Algorithm 1 Branch-and-cut Algorithm for Solving Problem (2)

Step 1: (Initialization) Let tolerance $\epsilon \geq 0$ for optimality. Define initial feasible region Ω as (5)

Step 2: (Branch and bound) Start (or continue) branch-and-bound algorithm to solve the relaxed problem:

$$\begin{aligned} \min\{\frac{\lambda}{|S|} \sum_{s \in S} (-\sum_{i \in I_s} P_i^s z_i^s + C_1 \sum_{j \in J} p_j + C_2 v) \\ + (1-\lambda) u : (\alpha, v, u, \boldsymbol{p}, \boldsymbol{z}, \boldsymbol{n}) \in \Omega\}. \end{aligned}$$

Step 3: (Termination criterion) Once a better feasible solution $(\widehat{\alpha}, \widehat{v}, \widehat{u}, \widehat{\boldsymbol{p}}, \widehat{\boldsymbol{z}}, \widehat{\boldsymbol{n}})$ in Ω is found, return the solution and current *MIP Gap*. If *MIP Gap* $\leq \epsilon$, stop the algorithm; otherwise go to **Step 4**.

Step 4: (Callback procedure) If the solution $(\widehat{\alpha}, \widehat{v}, \widehat{u}, \widehat{\boldsymbol{p}}, \widehat{\boldsymbol{z}}, \widehat{\boldsymbol{n}})$ violates the CVaR constraint (2b), generate the cutting plane with (6) to update the feasible region:

$$\Omega \leftarrow \Omega \cap \{(\alpha, v, u, \boldsymbol{p}, \boldsymbol{z}, \boldsymbol{n}) : (6)\},$$

and go to **Step 2** to continue branch-and-bound algorithm.

B. SCENARIO DECOMPOSITION ALGORITHM

In addition to the cutting-plane representation, we can also transform the nonlinear CVaR function into a linear one based on the lifting representation [25], [27]. Again, the primal problem is equivalently written in the lifting representation as follows.

$$\begin{aligned} \min \frac{\lambda}{|S|} \sum_{s \in S} \left(-\sum_{i \in I_s} P_i^s z_i^s + C_1 \sum_{j \in J} p_j + C_2 v \right) \\ + (1-\lambda) \left(\alpha + \frac{1}{(1-\beta)|S|} \sum_{s \in S} w^s \right) \end{aligned} \tag{7a}$$

subject to:

$$w^s \geq -\sum_{i\in I_s} P_i^s z_i^s + C_1 \sum_{j\in J} p_j + C_2 v - \alpha \tag{7b}$$

$$w^s \in \mathbb{R}_+ \tag{7c}$$

(1b)–(1l)

where $w^s, s \in S$ are the auxiliary decision variables. Hence, our formulation becomes a typical two-stage MILP model, which exhibits a block-angular structure and can be exploited in a decomposition fashion. One possible way of solution is to employ the scenario decomposition method, which is based on the Lagrangian decomposition method [28]. Carøe and Schultz [29] first applied scenario decomposition to stochastic integer programming so that the primal problem can be split into more manageable scenario subproblems. The main strategy behind this approach is to create copies of the first-stage variables.

To facilitate subsequent interpretation, we present problem (7) in the following compact notation.

$$Z = \min_{\boldsymbol{x},\boldsymbol{y}} \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x} + \frac{1}{|S|}\sum_{s\in S} (\boldsymbol{q}^s)^{\mathrm{T}} \boldsymbol{y}^s \tag{8a}$$

subject to:

$$A\boldsymbol{x} \leq \boldsymbol{b} \tag{8b}$$

$$T\boldsymbol{x} + W\boldsymbol{y}^s \leq \boldsymbol{h}^s, \forall s \in S \tag{8c}$$

$$\boldsymbol{x} \in \mathcal{X} \tag{8d}$$

$$\boldsymbol{y}^s \in \mathcal{Y}, \forall s \in S \tag{8e}$$

where $\boldsymbol{c}$, $\boldsymbol{b}$, $\boldsymbol{q}^s$, $s \in S$, and $\boldsymbol{h}^s$, $s \in S$ are the known vectors; A, T, and W are the known matrices; $\boldsymbol{x}$ and $\boldsymbol{y}^s$, $s \in S$ are the first-stage and second-stage decision variable vectors, respectively; and $\boldsymbol{y} = (\boldsymbol{y}^1, \boldsymbol{y}^2, \cdots, \boldsymbol{y}^{|S|})$. In our context, it is easy to transform the objective function (7a) to (8a). The set of constraints (8b) is used to represent constraints (1b) and (1c), while the set of constraints (8c) represents constraints (1d)–(1g) and (7b). Finally, the set $\mathcal{X}$ represents constraints (1h)–(1j), while the set $\mathcal{Y}$ denotes constraints (1k), (1l), and (7c).

By replicating the first-stage variables, we can consider the following equivalent formulation.

$$Z = \min_{\boldsymbol{x},\boldsymbol{y}} \frac{1}{|S|}\sum_{s\in S} \left(\boldsymbol{c}^{\mathrm{T}}\boldsymbol{x}^s + (\boldsymbol{q}^s)^{\mathrm{T}} \boldsymbol{y}^s\right) \tag{9a}$$

subject to:

$$A\boldsymbol{x}^s \leq \boldsymbol{b}, \forall s \in S \tag{9b}$$

$$T\boldsymbol{x}^s + W\boldsymbol{y}^s \leq \boldsymbol{h}^s, \forall s \in S \tag{9c}$$

$$\boldsymbol{x}^s - \bar{\boldsymbol{x}} = 0, \forall s \in S \tag{9d}$$

$$\boldsymbol{x}^s \in \mathcal{X}, \forall s \in S \tag{9e}$$

$$\boldsymbol{y}^s \in \mathcal{Y}, \forall s \in S \tag{9f}$$

$$\bar{\boldsymbol{x}} \in \mathbb{R}^n \tag{9g}$$

where $\boldsymbol{x} = (\boldsymbol{x}^1, \boldsymbol{x}^2, \cdots, \boldsymbol{x}^{|S|})$.

Equations (9d) are known as the nonanticipativity constraints. These constraints can be represented in various forms [30], [31]. By dualizing the nonanticipativity constraints (9d), one may obtain the Lagrangian relaxation of problem (10).

$$\begin{aligned} Z_{LR}(\boldsymbol{\mu}) = \min_{\boldsymbol{x},\boldsymbol{y},\bar{\boldsymbol{x}}} \frac{1}{|S|}\sum_{s\in S} \left(\boldsymbol{c}^{\mathrm{T}}\boldsymbol{x}^{\boldsymbol{s}} + (\boldsymbol{q}^{\boldsymbol{s}})^{\mathrm{T}} \boldsymbol{y}^s\right) \\ + \sum_{s\in S} (\boldsymbol{\mu}^s)^{\mathrm{T}} (\boldsymbol{x}^{\boldsymbol{s}} - \bar{\boldsymbol{x}}) \end{aligned} \tag{10}$$

subject to: (9b), (9c), and (9e)–(9g).

Since $\bar{\boldsymbol{x}}$ is unconstrained, we bound the Lagrangian with the condition $\sum_{s\in S} \boldsymbol{\mu}^s = 0$ and remove the term $\sum_{s\in S} (\boldsymbol{\mu}^s)^{\mathrm{T}} \bar{\boldsymbol{x}}$ in the Lagrangian function, which makes problem (10) separable. It is known that, for any given $\boldsymbol{\mu} = (\boldsymbol{\mu}^1, \boldsymbol{\mu}^2, \cdots, \boldsymbol{\mu}^{|S|})$, the Lagrangian relaxation (10) provides a valid lower bound on the problem (9) [32], i.e., $Z_{LR}(\mu) \leq Z$. To achieve such a bound, we can solve problem (10) by dividing it into many simpler subproblems. Note that $Z_{LR}(\mu) = \sum_{s\in S} Z_{LR}^s(\boldsymbol{\mu}^s)$, where $Z_{LR}^s(\boldsymbol{\mu}^s)$ is the objective value of the following subproblem.

$$Z_{LR}^s(\boldsymbol{\mu}^s) = \min_{\boldsymbol{x},\boldsymbol{y}} \frac{1}{|S|}\left(\boldsymbol{c}^{\mathrm{T}}\boldsymbol{x}^s + (\boldsymbol{q}^s)^{\mathrm{T}}\boldsymbol{y}^s\right) + (\boldsymbol{\mu}^s)^{\mathrm{T}} \boldsymbol{x}^s \tag{11}$$

subject to: (9b), (9c), and (9e)–(9g), for a given s in S.

To find the best lower bound, we define the Lagrangian dual problem.

$$Z_{LD} = \max_{\boldsymbol{\mu}} \frac{1}{|S|}\sum_{s\in S} Z_{LR}^s \tag{12a}$$

subject to:

$$\sum_{s\in S} \boldsymbol{\mu}^s = 0 \tag{12b}$$

$$\boldsymbol{\mu}^s \in \mathbb{R}^n, \forall s \in S \tag{12c}$$

This dual problem (12) is a concave, nonsmooth optimization problem, which is commonly solved using methods based on subgradients or cutting planes [31]. In this study, a cutting-plane method is used to handle the dual problem. **Algorithm 2** is a stabilized cutting-plane approach with a trust region [33], [34]. Hiriart-Urrut and Lemaréchal [33] proved the convergence of this approach. In this algorithm, the master problem and local problem, respectively, yield an upper bound (Z_{CP}) and lower bound ($Z_{LR}(\mu) = \sum_{s\in S} Z_{LR}^s(\boldsymbol{\mu}^s)$) of Z_{LD}. By iterating the upper and lower bounds in turn, the algorithm terminates when the relative gap between them is less than a given optimality tolerance.

Algorithm 2 solves the Lagrangian dual problem (12) and derives the lower bound of the primal problem (8). Next, we will introduce a simple heuristic to determine the upper bound. The upper bound can be calculated by substituting the feasible first-stage solutions $\boldsymbol{x}^s$ into the primal problem (8). In this study, we estimated the feasible first-stage candidates by using the average value of the scenario solution $\boldsymbol{x}^s$, $s \in S$ and some rounding heuristic to satisfy the

Algorithm 2 Trust-region Cutting-plane Algorithm for Solving Lagrangian Dual Problem (12)

Step 1: (Initialization) Set the tolerance $\epsilon \geq 0$, iteration count $k = 1$, initial stability center $\bar{\boldsymbol{\mu}}^s = \boldsymbol{\mu}_k^s = \mathbf{0}$, $s \in S$, ascent coefficient $\omega \in (0,1)$, and initial trust-region $\tau_k \geq 0$, solve subproblem (11) for all s in S, and save $Z_{LR}^s(\boldsymbol{\mu}_k^s)$ and $\boldsymbol{x}_k^s$.

Step 2: (Master problem) Solve the following problem

$$
\begin{aligned}
Z_{CP}^k = \max_{\theta,\boldsymbol{\mu}} & \sum_{s\in S} \theta^s \\
\text{subject to:} & \\
\theta^s \leq & Z_{LR}^s(\boldsymbol{\mu}_{k'}^s) + (\boldsymbol{x}_{k'}^s)^{\mathrm{T}}(\boldsymbol{\mu}^s - \boldsymbol{\mu}_{k'}^s), \\
& \forall s \in S, k' \in \{1,2,\cdots,k\} \\
& |\boldsymbol{\mu}^s - \bar{\boldsymbol{\mu}}^s| \leq \tau_k, \forall s \in S \\
& \sum_{s\in S} \boldsymbol{\mu}^s = 0 \\
& \boldsymbol{\mu}^s \in \mathbb{R}^n, \forall s \in S \\
& \theta^s \in \mathbb{R}, \forall s \in S
\end{aligned}
$$

to obtain $\left(Z_{CP}^k, \boldsymbol{\mu}_{k+1}^s\right)$, and compute

$$\delta_k := Z_{CP}^k - \sum_{s\in S} Z_{LR}^s(\bar{\boldsymbol{\mu}}^s).$$

Step 3: (Termination criterion) If

$$\frac{\delta_k}{1 + \sum_{s\in S} Z_{LR}^s(\bar{\boldsymbol{\mu}}^s)} \leq \epsilon,$$

stop the algorithm, otherwise go to **Step 4**.

Step 4: (Local problem) For all s in S, solve $|S|$ subproblems (11) to obtain the next points $\boldsymbol{x}_{k+1}^s$ and $Z_{LR}^s(\boldsymbol{\mu}_{k+1}^s)$.

Step 5: (Center update) If

$$\sum_{s\in S} Z_{LR}^s\left(\boldsymbol{\mu}_{k+1}^s\right) \geq \sum_{s\in S} Z_{LR}^s(\bar{\boldsymbol{\mu}}^s) + \omega\delta_k,$$

update stability center $\bar{\boldsymbol{\mu}}^s = \boldsymbol{\mu}_{k+1}^s, \forall s \in S$, otherwise leave the center unchanged.

Step 6: (Trust region update) Compute the ratio

$$\rho := \frac{\sum_{s\in S} Z_{LR}^s(\boldsymbol{\mu}_{k+1}^s) - \sum_{s\in S} Z_{LR}^s(\bar{\boldsymbol{\mu}}^s)}{\delta_k}.$$

If $\rho = 1$, then $\tau_{k+1} = 1.5\tau_k$ and if $\rho < 0$, then $\tau_{k+1} = 0.8\tau_k$.

Step 7: (Iteration update) Set $k = k+1$, go back to **Step 2** .

integrality restriction ($\boldsymbol{x} = \text{round}(\sum_{s\in S} \boldsymbol{x}^s/|S|)$). When the first-stage variables are known, the primal problem can also be decomposed into $|S|$ subproblems to compute the upper bound of the primal problem (8).

Algorithm 3 shows the whole scenario decomposition method combining **Algorithm 2** and the method of computing the upper bound. In detail, it only adds the upper bound update (**Algorithm 3 Step 3**) and a new termination criterion

Algorithm 3 Scenario Decomposition Method for Solving Primal Problem (8)

Step 1: (Initialization) Set the tolerance $\epsilon_{CP} \geq 0$, $\epsilon_{DG} \geq 0$ iteration count $k = 1$, initial stability center $\bar{\boldsymbol{\mu}}^s = \boldsymbol{\mu}_k^s = \mathbf{0}$, $s \in S$, an ascent coefficient $\omega \in (0,1)$, and initial trust-region $\tau_k \geq 0$. Set upper bound $UB = +\infty$, lower bound $LB = -\infty$

Step 2: (Lower bound update) solve subproblem (11) for all s in S to obtain $Z_{LR}^s(\boldsymbol{\mu}_k^s)$ and $\boldsymbol{x}_k^s$; clearly, $Z_{LR}(\mu_k) = \sum_{s\in S} Z_{LR}^s(\boldsymbol{\mu}^s)$. If $Z_{LR}(\boldsymbol{\mu}_k) > LB$, then $LB = Z_{LR}(\boldsymbol{\mu}_k)$.

Step 3: (Upper bound update) Generate the first-stage variable with $\widehat{\boldsymbol{x}}_k = \boldsymbol{x}_k^s/|S|$. For integer variables, $\widehat{\boldsymbol{x}}_k = \text{round}\boldsymbol{x}_k^s/|S|)$. Then, fix $x = \widehat{\boldsymbol{x}}_k$ in primal problem (8) to compute the optimal Z. If $Z < UB$, then $UB = Z$.

Step 4: (Termination criterion) If $UB - LB \leq \epsilon_{DG}$, stop the algorithm; otherwise go to **Step 5**.

Step 5: (Cut generation) Execute **Algorithm 2** Step 2 to obtain $\left(Z_{CP}^k, \boldsymbol{\mu}_{k+1}^s, \delta_k\right)$

Step 6: (Termination criterion) If

$$\frac{\delta_k}{1 + \sum_{s\in S} Z_{LR}^s(\bar{\boldsymbol{\mu}}^s)} \leq \epsilon_{CP},$$

stop the algorithm, otherwise go to **Step 7**.

Step 7: (Center and trust region update) Perform **Steps 1–3** again to obtain $Z_{LR}^s(\boldsymbol{\mu}_k^s)$, $\boldsymbol{x}_k^s$, LB, and UB, then update the stability center and trust region with **Step 5** and **6** of **Algorithm 2**.

Step 8: (Iteration update) Set $k = k+1$, go back to **Step 5**.

(**Algorithm 3 Step 4**) between **Steps 1** and **2** of **Algorithm 2**.

Because of the block structure, the scenario decomposition method can split up the primal problem (8) into $|S|$ scenario subproblems. In addition, these scenarios can be grouped together into larger blocks to help reduce the gap [29] while maintaining the block structure; thereby, we will rewrite problem (9) into problem (13).

$$Z = \min_{\boldsymbol{x},\boldsymbol{y}} \frac{1}{|L|} \sum_{l\in L} \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x}^l + \frac{1}{|S|} \sum_{l\in L} \sum_{s\in S_l} (\boldsymbol{q}^s)^{\mathrm{T}}\boldsymbol{y}^s \tag{13a}$$

subject to:

$$A\boldsymbol{x}^l \leq \boldsymbol{b}, \forall l \in L \tag{13b}$$

$$T\boldsymbol{x}^l + W\boldsymbol{y}^s \leq \boldsymbol{h}^s, \forall l \in L, s \in S_l \tag{13c}$$

$$\boldsymbol{x}^l - \bar{\boldsymbol{x}} = 0, \forall l \in L \tag{13d}$$

$$\boldsymbol{x}^l \in \mathcal{X}, \forall l \in L \tag{13e}$$

$$\boldsymbol{y}^s \in \mathcal{Y}, \forall l \in L, s \in S_l \tag{13f}$$

$$\bar{\boldsymbol{x}} \in \mathbb{R}^n \tag{13g}$$

where L is the set of blocks and $l \in L$, S_l is the set of scenarios in block l. $|L|$ is chosen to be a suitable divisor of $|S|$. Similarly, we can solve problem (13) by the scenario decomposition method. According to our preliminary experiments, instead of formulating subproblems for each scenario,

we can split the scenarios into several large blocks in the order of the scenario, which leads to a better solution.

V. COMPUTATIONAL EXPERIMENTS

We conducted experiments based on historical data of the Ha:mo RIDE carsharing system in Toyota city, Japan collected from April 1st, 2016 to March 31st, 2017. Ha:mo RIDE is a station-based one-way system. Considering the accessibility of data, we utilized the data from the existing carsharing system directly to generate scenario demand data and assumed the sites of the current stations to be potential location sites. Note that the formulation proposed in Section III is not affected by these treatments or by the other methods used to obtain the demand data and potential locations (e.g., [4], [7]). The model was implemented using Gurobi Optimizer 9.0 in the Python environment on an Intel i7-8700 CPU with 12 cores and 16GB of RAM.

A. PARAMETER SETTINGS

There were 55 stations with different capacities in the Ha:mo RIDE system. As mentioned above, the sites of these stations were assumed to be the potential location sites and the current station capacities were regarded as maximum numbers of parking spaces that can be set. The operating time of the system was from 6:00 to 24:00, which in the experiments was divided into multiple 5-minute intervals.

An important part of the parameters in the proposed model is the scenario demand data. Instead of predicting the potential demand with forecast methods like regression prediction, we generated the demand by using a Poisson distribution due to the limited data, which is the same as [17]. The average origin-destination (OD) matrices for the same hours were calculated from the historical data for different days. The number of OD trips per hour was then evenly distributed over the time intervals, which determines the parameter in the Poisson distribution. In such a way, we generated a random number of possible trips departing from each station for each time interval. Furthermore, the arrival time was determined by adding the trip duration to the departure time. The trip duration for each OD pair varied every hour, which was obtained from the Google Maps Distance Matrix API. We based the trip price on the trip duration: a fare of ¥200 for up to 10 minutes and ¥20 per minute after that. Using this procedure, we repeatedly generated different scenario demands.

The remaining parameters are mostly related to the costs. Besides the station cost, other costs are constant ones, including the station and vehicle operating costs and the cost of purchasing vehicles. Because the station cost involves the land cost, which varies according to the local conditions, we collected the land cost data for Toyota City from the National Land Price Map, Japan [35]. The constant parameters are summarized in Table 1. Note that some parameters have been modified to ensure the system can make a profit.

TABLE 1. Values of constant parameters used in the model.

Parameters	Value
Operating cost per parking space per scenario (C_1)	¥100
Operating cost per vehicle per scenario(C_2)	¥200
Cost for purchasing a vehicle (F)	¥879000
Available budget (B)	¥200 million
Number of scenarios ($\|S\|$)	1000

B. OPTIMIZATION RESULTS

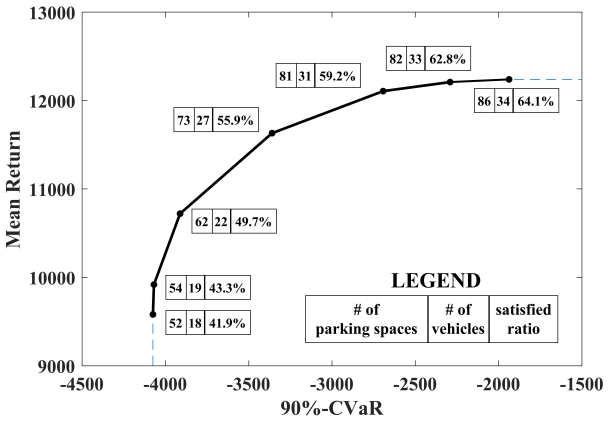
With the given parameters, we solved the strategic decision model directly with Gurobi and generated efficient frontiers for the 90%, 95%, and 99% confidence levels by using different weight values , as shown in Fig. 1. This figure also shows optimal results including the total number of parking spaces, the number of vehicles used, and the demand satisfaction rate for different weight values. From the efficient frontiers, it is obvious that high returns are accompanied by high risks. To achieve higher returns, the main approach appears to be increasing the number of parking spaces or vehicles, which improves the satisfied demand ratio. By considering different confidence levels, we found that a higher confidence level quantifies more serious risk and reduces the return to a certain degree. Furthermore, it seems sufficient to set 86 parking spaces and purchase 34 vehicles in the carsharing system, since more parking spaces or vehicles increase risk, not return.

Fig. 2 illustrates the optimal location and capacity of stations for different values of λ, given β = 95%. The size of the circle represents the number of parking spaces. The individual figures show the optimal number of stations, number of parking spaces, and required fleet size in the system at the upper right corner. Generally, stations equipped with more parking spaces are mainly located at the spots with high demand, such as at Toyota factories and railway stations. However, these hot spots are easily affected by the weight values; that is, the number of parking spaces will increase when more attention is paid to the return. In comparison, the solutions seem more robust for some small stations. There are 13 common small stations, marked in red circles, where the location and capacity of stations are the same.

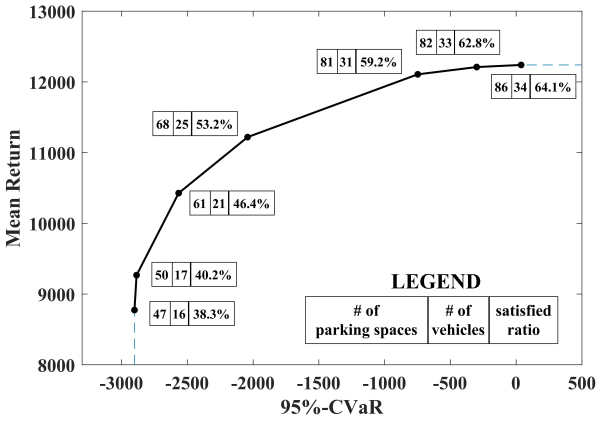
C. OUT-OF-SAMPLE PERFORMANCE OF THE RISK-AVERSE MODEL

As mentioned above, most of the previous studies maximized the expectation value of returns in the optimization model without any consideration of risk. In this subsection, we develop an evaluation method that is similar to the training and testing method used in the field of machine learning to verify whether introducing risk has benefits, i.e., whether the strategic decisions from risk-averse model are better. As can be seen in Fig. 3, the proposed evaluation process consists of two important parts. In the first part (training part), we set different weight values in the strategic model to optimize the corresponding strategic decisions. In the second part (testing part), we render these decisions as additional

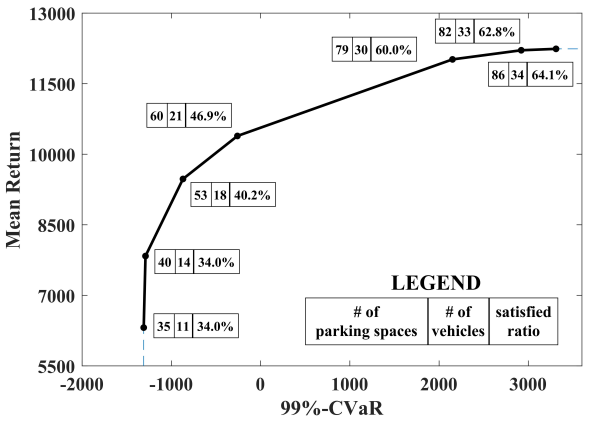
This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3084287, IEEE Access



(a)



(b)



(c)

FIGURE 1. Efficient frontiers of mean return and CVaR: (a) $\beta = 90\%$; (b) $\beta = 95\%$; (c) $\beta = 99\%$.

constraints in the strategic model and meanwhile input test demand data into the model. After optimizing other variables according to the determined variables associated with the strategic decisions, the mean returns in the objective function are compared to further inspect the strategic decisions.

To examine the impact of introducing the risk term on the strategic decisions, the weight values (λ) used to generate the

TABLE 2. Mean return on test data for different λ when $\beta = 90\%$.

Test data	$\lambda = 0$	$0 < \lambda < 1$							$\lambda = 1$
		0.001	0.1	0.3	0.5	0.7	0.9	0.999	
1	5335	9506	9830	10628	11485	11920	11990	**12022**	**12022**
2	4140	8593	8853	9511	10135	10388	**10397**	10340	10340
3	3980	8296	8514	9033	9566	**9790**	9734	9708	9708
4	3190	7956	8210	8681	9122	**9217**	9154	9138	9138
5	3191	7530	7759	8236	8524	**8567**	8490	8350	8350
6	3065	6845	7011	7331	**7394**	7328	7142	7019	7019
7	3139	6692	6794	7038	**7155**	6944	6720	6583	6583
8	3589	6396	6494	6670	**6684**	6476	6238	6078	6078
9	2659	5399	5460	**5532**	5178	4709	4403	4207	4207
10	2792	4411	**4461**	4359	3815	3161	2764	2459	2459
Average	3508	7162	7339	7702	**7906**	7850	7703	7590	7590

TABLE 3. Mean return on test data for different λ when $\beta = 95\%$.

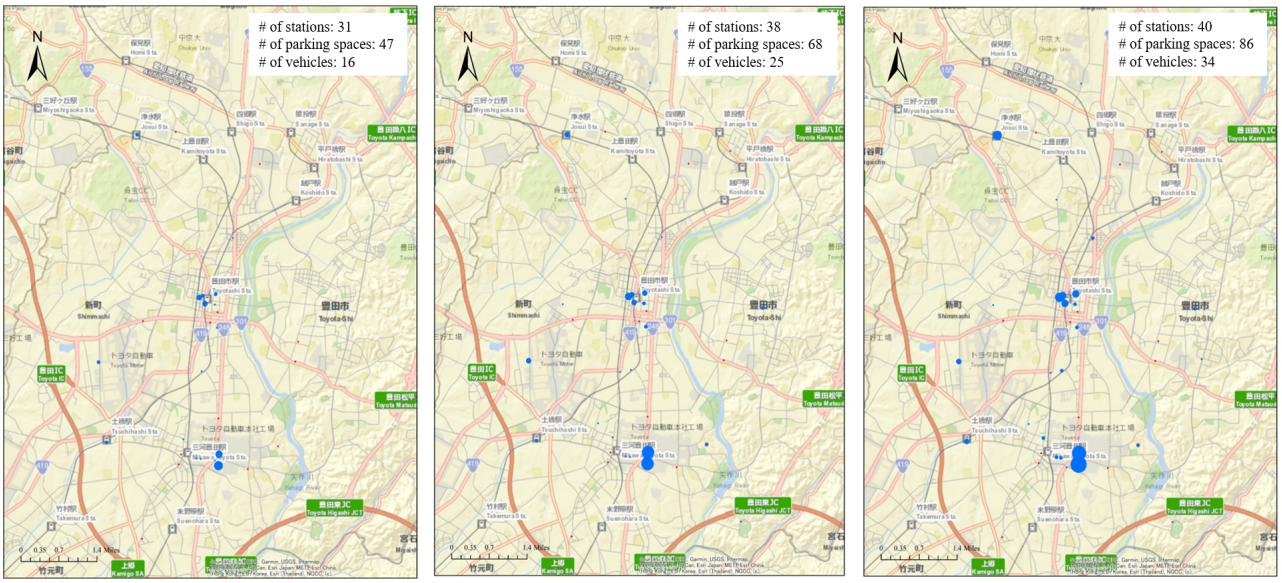
Test data	$\lambda = 0$	$0 < \lambda < 1$							$\lambda = 1$
		0.001	0.1	0.3	0.5	0.7	0.9	0.999	
1	3881	8762	9200	10310	11088	11920	11990	**12022**	**12022**
2	2889	7887	8336	9303	9890	10388	**10397**	10340	10340
3	3604	7632	8048	8795	9322	**9790**	9734	9708	9708
4	2655	7365	7735	8509	8938	**9217**	9154	9138	9138
5	2759	6871	7313	8026	8408	**8567**	8490	8350	8350
6	2790	6272	6646	7213	**7377**	7328	7142	7019	7019
7	2727	6172	6531	6971	**7079**	6944	6720	6583	6583
8	2805	5913	6241	6630	**6660**	6476	6238	6078	6078
9	2598	5071	5321	**5537**	5376	4709	4403	4207	4207
10	2215	4163	**4375**	4374	4063	3161	2764	2459	2459
Average	2892	6611	6975	7567	7820	**7850**	7703	7590	7590

efficient frontier are divided into three categories: i) $\lambda = 0$, i.e., only risk is considered; ii) $0 < \lambda < 1$, i.e., both return and risk are considered; iii) $\lambda = 1$, i.e., only return is considered. Given different confidence levels β, the strategic decisions with respect to different λ will be evaluated by the indicator, mean return, on diverse test demand data. We generated 10 sets of test demand data that followed a Poisson distribution. The test results for $\beta = 90\%, 95\%$, and 99% are reported in Tables 2, 3, and 4, respectively.

For each set of test data, the maximal return is marked in bold. In Table 2, it is interesting that, for most of the test data, the maximal returns are obtained when $0 < \lambda < 1$ rather than $\lambda = 1$. Similar observations apply to Tables 3 and 4. Strategic decisions based on a single criterion are more likely to cause poor performance under demand uncertainty, which indicates the necessity of introducing the risk term. Weighting the return against risk, the carsharing system operator may earn higher returns in the future. In addition, we also find that appropriate attention should be paid to risk (i.e., choosing a suitably smaller λ) if the demand data results in lower returns. Take test data 1 and 10 in Table 2 as examples. The maximal returns are obtained when $\lambda = 0.1$ for test data 10 and $\lambda = 0.999$ or 1 for test data 1, and it can be seen that the return from test data 10 is lower than the return from test data 1.

D. COMPARISON OF PROPOSED SOLUTION METHODS

To assess the efficiency of our algorithms, we conducted computational experiments comparing the two proposed al-



(a) (b) (c)

FIGURE 2. Optimal station locations and capacities for $\beta = 95\%$: (a) $\lambda = 0$; (b) $\lambda = 0.5$; (c) $\lambda = 1$.

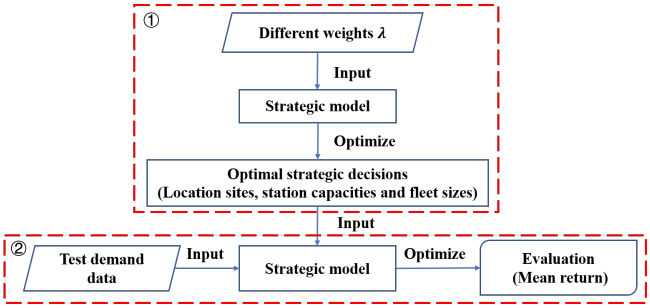


FIGURE 3. Scheme to evaluate the model with risk term.

TABLE 4. Mean return on test data for different λ when $\beta = 99\%$.

Test data	$\lambda = 0$	$0 < \lambda < 1$							$\lambda = 1$
		0.001	0.1	0.3	0.5	0.7	0.9	0.999	
1	1757	6254	7755	9391	10289	11849	11990	**12022**	**12022**
2	1937	5702	7041	8519	9206	10330	**10397**	10340	10340
3	2844	5569	6919	8166	8806	**9766**	9734	9708	9708
4	1747	5372	6656	7875	8438	**9220**	9154	9138	9138
5	1754	5065	6144	7476	7985	**8542**	8490	8350	8350
6	1033	4716	5716	6791	7218	**7334**	7142	7019	7019
7	388	4660	5574	6537	6962	**7010**	6720	6583	6583
8	1738	4441	5341	6242	**6589**	6474	6238	6078	6078
9	1036	3892	4547	5345	**5457**	4776	4403	4207	4207
10	1612	3343	3785	**4384**	4309	3263	2764	2459	2459
Average	1585	4901	5948	7073	7526	**7856**	7703	7590	7590

gorithms with direct usage of the Gurobi optimization solver and Benders decomposition-based algorithm used in [23]. For each parameter set (λ, β, and $|S|$), we designed three experiments by generating different trip demands in each scenario, and we evaluated the average performance of these experiments. We set the weight parameter, $\lambda \in \{0.1, 0.5, 0.9\}$, the confidence level parameter, $\beta \in \{90\%, 95\%, 99\%\}$, and the number of scenarios, $|S| \in \{200, 1000, 2000\}$. When using Gurobi directly and employing the branch-and-bound algorithm, both optimality tolerances take the default value, $\epsilon = 10^{-4}$. In the scenario decomposition algorithm, the optimality tolerances ϵ_{CP} and ϵ_{DG} are 10^{-4} and 300, respectively. For all methods, the time limits are set to be 7200s.

In Tables 5, 6, and 7, "Direct" means direct usage of Gurobi, "B&C" is the branch-and-cut algorithm, "SD" represents the scenario decomposition method, and "BD" means the Benders decomposition-based algorithm introduced in [23]. On large-scale problems, the algorithms usually terminated because they ran out of memory. Accordingly, "OM(*)" means that the algorithm experienced a memory shortage, where "*" is the number of memory shortages out of three. Similarly, "OT(*)" depicts the computation was terminated due to time limit, where "*" is the number out of three. Moreover, "Gap (%)" is the relative optimality gap.

Tables 5, 6, and 7 illustrate the results of solving the problems with different numbers of scenarios, mainly including the gaps and time. As can be seen in Tables 5 and 6, both direct method and B&C algorithm can provide optimal values with gaps being less than 10^{-4}. For some problems with 200 scenarios, the B&C algorithm takes a little more time than the direct approach. However, when the number of scenarios is 1000, the B&C algorithm shows more favorable results; its computing time was reduced by 51% on average in comparison with Gurobi. Turning now to the SD algorithm, we see that the algorithm is faster than both the direct approach and the B&C algorithm and the resulting optimality gaps are mostly within 3%. Although the SD algorithm can provide tight lower bounds with the trust-region cutting-plane method, a simple heuristic to estimate the upper bounds sometimes may cause weak solutions and large gaps (e.g., $\lambda = 0.1$, $\beta = 99\%$), which can be further improved. In comparison, BD performs worse than all other methods. For

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3084287, IEEE Access

TABLE 5. Results of solving problem with 200 scenarios.

λ	β	Gap (%)				Time (s)			
		Direct	B&C	SD	BD	Direct	B&C	SD	BD
	90%			5.33		58.0	64.7	33.2	
0.1	95%			12.35		46.3	71.2	40.3	
	99%			24.62		150.0	231.0	88.7	
	90%			1.73		59.1	50.8	33.3	
0.5	95%		≤ 0.01	2.60	> 100	68.5	50.8	29.6	OT(3)
	99%			3.28		82.5	86.5	36.9	
	90%			0.35		64.8	37.1	23.4	
0.9	95%			0.46		57.3	43.2	23.5	
	99%			0.81		47.5	50.9	21.0	

TABLE 6. Results of solving problem with 1000 scenarios.

λ	β	Gap (%)				Time (s)			
		Direct	B&C	SD	BD	Direct	B&C	SD	BD
	90%			1.08		3587.3	2312.3	1132.0	
0.1	95%			2.48		4035.3	2884.4	1025.0	
	99%			17.16		4137.8	2966.4	1430.4	
	90%			0.29		5202.5	2057.7	1147.6	
0.5	95%		≤ 0.01	0.97	>100	3595.5	1597.8	1063.3	OT(3)
	99%			2.91		3232.4	1958.3	1059.4	
	90%			0.06		3044.1	907.5	949.5	
0.9	95%			0.06		3582.9	1003.1	878.4	
	99%			0.22		3299.2	1144.2	867.1	

all problems with 200 scenarios or 1000 scenarios, we did not obtain good solutions within the specified time by using the BD algorithm. With the increase of the number of iterations, the master problem in the BD algorithm becomes a larger MILP gradually and most of the time is spent on solving the master problem. However, in the SD algorithm, we substitute a heuristic solution (average value) into the prime problem to get the solution without solving such an MILP. In short, both B&C algorithm and SD algorithm are effective when solving the problem with fewer scenarios, and one can choose the B&C algorithm and SD algorithm according to the need for a smaller gap or for higher speed .

Now, let us look at Table 7, which indicates the advantage and efficiency of the SD algorithm for solving large-scale problem. The core of the B&C algorithm is replacing the CVaR constraint with the cutting-plane representation, so a problem remains is that the other linear constraints still depend on the number of scenarios, i.e., a much greater number of scenarios may result in difficulty finding a good solution. The table indicates that the direct approach and B&C algorithm could not solve the problems with 2000 scenarios and all experiments terminated due to memory shortage. BD method could not find a feasible solution for our model within the given time. Nevertheless, the SD algorithm can compensate for hardware defects by consuming more (but still tolerable) computation time and obtain favorable solutions. Even for the problems with $\lambda = 0.1$ and $\beta = 99\%$, the gap presents a downward trend compared with the values in Tables 5 and 6. These illustrate the superiority of SD on large-scale problems.

TABLE 7. Results of solving problem with 2000 scenarios.

λ	β	Gap (%)				Time (s)			
		Direct	B&C	SD	BD	Direct	B&C	SD	BD
	90%			1.94				1804.6	
0.1	95%			5.57				OT(3)	
	99%			13.96				4216.5	
	90%			0.60				1924.4	
0.5	95%		-	1.14	-		OM(3)	2243.5	OT(3)
	99%			4.52				OT(3)	
	90%			0.20				3739.5	
0.9	95%			0.21				2957.9	
	99%			0.45				6666.1	

VI. CONCLUSION

We proposed a two-stage stochastic risk-averse MINLP model to optimize the strategic decisions in one-way station-based carsharing systems operating under demand uncertainty. In the model, the optimal location, capacity of stations, and fleet size can be determined at the same time. In addition to the expected return that is a common optimization objective, the risk measure CVaR is incorporated into the model so that the operator can examine the trade-off between return and risk. Since we aimed to solve the problem efficiently, we developed two methods, a branch-and-cut algorithm and a scenario decomposition algorithm, by converting the primal problem into two different equivalent problems.

Using the historical demand data, we generated scenario demands that followed a Poisson distribution and conducted computational experiments. By solving the problems with different weight values, we obtained efficient frontiers, which revealed a positive correlation between return and CVaR. From the efficient frontiers, we found that building more parking spaces or preparing more vehicles can improve the return and the satisfied demand ratio, but that they cause more risk. The stations equipped with more parking spaces are suggested to be set at the spots with high demand, such as at Toyota factories and railway stations. Additionally, we evaluated the advantages of the proposed model with the risk term. The results show that it is better to consider both the return and risk in the objective function so that the strategic decisions may lead to a higher return. When comparing the solution methods, the algorithm developed by Lu *et al.* [23] has unsatisfactory performance on our model. By contrast, both branch-and-cut and scenario decomposition algorithms are effective at solving small- and medium-scale problems. More importantly, the scenario decomposition can deal with large-scale problems that cannot be solved by the direct approach or by the branch-and-cut algorithm.

An interesting challenge would be to develop more complicated variations of our model with constraints such as energy consumption, vehicle relocations, or relays, which will relax our assumptions and make the model more realistic. In particular, given the current scale of the proposed model, relaxation tricks might be needed to handle these constraints. On the other hand, from a computational perspective, we can improve the heuristic method for determining upper bounds in scenario decomposition to make the algorithm

more sophisticated.

.

REFERENCES

[1] R. Katzev, "Car sharing: A new approach to urban transportation problems," *Anal. Soc. Issues Public Policy*, vol. 3, no. 1, pp. 65–86, 2003.

[2] M. Barth and S. A. Shaheen, "Shared-use vehicle systems: Framework for classifying carsharing, station cars, and combined approaches," *Transp. Res. Rec. J. Transp. Res. Board*, vol. 1791, pp. 105–112, Jan. 2002. [Online]. Available: http://journals.sagepub.com/doi/10.3141/1791-16

[3] G. H. de Almeida Correia, D. R. Jorge, and D. M. Antunes, "The added value of accounting for users flexibility and information on the potential of a station-based one-way car-sharing system: An application in Lisbon, Portugal," *J. Intell. Transp. Syst. Technol. Planning, Oper.*, vol. 18, pp. 299–308, July 2014.

[4] B. Boyaci, K. G. Zografos, and N. Geroliminis, "An optimization framework for the development of efficient one-way car-sharing systems," *Eur. J. Oper. Res.*, vol. 240, pp. 718–733, Feb. 2015.

[5] S. Illgen and M. Höck, "Literature review of the vehicle relocation problem in one-way car sharing networks," *Transp. Res. Part B Methodol.*, vol. 120, pp. 193–204, Feb. 2019.

[6] G. Brandstätter, M. Kahr, and M. Leitner, "Determining optimal locations for charging stations of electric car-sharing systems under stochastic demand," *Transp. Res. Part B Methodol.*, vol. 104, pp. 17–35, Oct. 2017.

[7] H. Çalık and B. Fortz, "A benders decomposition method for locating stations in a one-way electric car sharing system under demand uncertainty," *Transp. Res. Part B Methodol.*, vol. 125, pp.121–150, July 2019.

[8] N. Noyan, "Risk-averse two-stage stochastic programming with an application to disaster management," *Comput. Oper. Res.*, vol. 39, pp. 541–559, 2012. [Online]. Available: http://dx.doi.org/10.1016/j.cor.2011.03.017

[9] G. Brandstätter, C. Gambella, M. Leitner, E. Malaguti, F. Masini, J. Puchinger, M. Ruthmair, and D. Vigo, "Overview of optimization problems in electric car-sharing system design and management," in *Dynamic perspectives on managerial decision making*. Cham, Switzerland: Springer, Cham, 2016, pp. 441–471.

[10] D. Gavalas, C. Konstantopoulos, and G. Pantziou, "Design and management of vehicle-sharing systems: A survey of algorithmic approaches," in *Smart Cities and Homes*. Boston, MA, USA: Morgan Kaufmann, 2016, pp. 261–289.

[11] B. Boyacı, K. G. Zografos, and N. Geroliminis, "An integrated optimization-simulation framework for vehicle and personnel relocations of electric carsharing systems with reservations, " *Transp. Res. Part B Methodol.*, vol. 95, pp. 214–237, Jan. 2017.

[12] M. Bruglieri, A. Colorni, and A. Luè, "The relocation problem for the one-way electric vehicle sharing," *Networks*, vol. 64, pp. 292–305, Nov. 2014. [Online]. Available: http://doi.wiley.com/10.1002/net.21585

[13] W. D. Fan, "Management of dynamic cehicle allocation for carsharing cystems," *Transp. Res. Rec. J. Transp. Res. Board*, vol. 2359, pp. 51–58, Jan. 2013. [Online]. Available: http://journals.sagepub.com/doi/10.3141/2359-07

[14] D. Jorge, G. H. de Almeida Correia, and C. Barnhart, "Comparing optimal relocation operations with simulated relocation policies in one-way carsharing systems," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, pp. 1667–1675, 2014.

[15] K. Huang, K. An, J. Rich, and W. Ma, "Vehicle relocation in one-way station-based electric carsharing systems: A comparative study of operator-based and user-based methods," *Transp. Res. Part E Logist. Transp. Rev.*, vol. 142, p. 102081, Oct. 2020.

[16] R. Nair and E. Miller-Hooks, "Fleet management for vehicle sharing operations," *Transp. Sci.*, vol. 45, pp. 524–540, Dec. 2011. [Online]. Available: https://pubsonline.informs.org/doi/abs/10.1287/trsc.1100.0347

[17] M. Yamada, M. Kimura, N. Takahashi, and A. Yoshise, "Optimization-based analysis of last-mile one-way mobility sharing," Department of Policy and Planning Sciences, University of Tsukuba, Tsukuba, Japan, *Discussion Paper Series*, no.1353, 2018. [Online]. Available: http://infoshako.sk.tsukuba.ac.jp/ databank/pdf/1353.pdf

[18] M. Zhao, X. Li, J. Yin, J. Cui, L. Yang, and S. An, "An integrated framework for electric vehicle rebalancing and staff relocation in one-way carsharing systems: Model formulation and lagrangian relaxation-based solution approach," *Transp. Res. Part B Methodol.*, vol. 117, pp. 542–572, Nov. 2018.

[19] M. S. Daskin, *Network and Discrete Location: Models, Algorithms, and Applications, Second Edition*, Hoboken, NJ, USA: John Wiley & Sons, Inc., 2013. [Online]. Available: http://doi.wiley.com/10.1002/9781118537015

[20] G. H. de Almeida Correia and A. P. Antunes, "Optimization approach to depot location and trip selection in one-way carsharing systems," *Transp. Res. Part E Logist. Transp. Rev.*, vol. 48, pp. 233–247, Jan. 2012.

[21] D. Jorge, G. H. de Almeida Correia, and C. Barnhart, "Testing the validity of the MIP approach for locating carsharing stations in one-way systems," *Procedia - Soc. Behav. Sci.*, vol. 54, pp. 138–148, Oct. 2012.

[22] K. Huang, G. H. de Almeida Correia, and K. An, "Solving the station-based one-way carsharing network planning problem with relocations and non-linear demand," *Transp. Res. Part C Emerg. Technol.*, vol. 90, pp. 1–17, May 2018.

[23] M. Lu, Z. Chen, and S. Shen, "Optimizing the profitability and quality of service in carshare systems under demand uncertainty,"*Manuf. Serv. Oper. Manag.*, vol. 20, no. 2, pp. 162–180, 2018

[24] K. Shimazaki, M. Kuwahara, A. Yoshioka, Y. Homma, M. Yamada, and A. Matsui, "Development of a simulator for one-way ev sharing service," Presented at 20th ITS World Congress, ITS Japan.

[25] Y. Takano, K. Nanjo, N. Sukegawa, and S. Mizuno, "Cutting plane algorithms for mean-cvar portfolio optimization with nonconvex transaction costs," *Comput. Manag. Sci.*, vol. 12, pp. 319–340, May 2014. [Online]. Available: https://link.springer.com/article/10.1007/s10287-014-0209-7

[26] A. Künzi-Bay and J. Mayer, "Computational aspects of minimizing conditional value-at-risk," *Comput. Manag. Sci.*, vol. 3, pp. 3–27, 2006. [Online]. Available:https://doi.org/10.1007/s10287-005-0042-0

[27] R. T. Rockafellar and S. Uryasev, "Optimization of conditional value-at-risk," *J. Risk*, vol. 2, pp. 21–41, 2000.

[28] M. L. Fisher, "Applications oriented guide to lagrangian relaxation," *Interfaces*, vol. 15, pp. 10–21, Apr. 1985. [Online]. Available: https://pubsonline.informs.org/doi/abs/10.1287/inte.15.2.10

[29] C. C. CarøE, and R. Schultz, "Dual decomposition in stochastic integer programming," *Oper. Res. Lett.*, vol. 24, no. 1–2, pp. 37-45, Feb. 1999.

[30] M. Lubin, K. Martin, C. G. Petra, and B. Sandikçi, "On parallelizing dual decomposition in stochastic integer programming," *Oper. Res. Lett.*, vol. 41, pp. 252–258, May 2013.

[31] F. Oliveira, V. Gupta, S. Hamacher, and I. E. Grossmann, "A Lagrangean decomposition approach for oil supply chain investment planning under uncertainty with risk considerations," *Comput. Chem. Eng.*, vol. 50, pp. 184–195, Mar. 2013.

[32] A. M. Geoffrion, "Lagrangean relaxation for integer programming," in *Mathematical Programming Studies, vol 2 Approaches to Integer Programming. M. L. Balinski, Ed*. Berlin, Heidelberg, Germany: Springer, 1974.

[33] J.-B. Hiriart-Urruty and C. Lemarechal. *Convex analysis and minimization algorithms II – Advanced theory and bundle methods*. Berlin, Heidelberg, Germany: Springer, 1993.

[34] B. R. Knudsen, I. E. Grossmann, B. Foss, and A. R. Conn, "Lagrangian relaxation based decomposition for well scheduling in shale-gas systems," *Comput. Chem. Eng.*, vol. 63, pp. 234–249, Apr. 2014.

[35] Research Center for Property Assessment System, National Land Price Map, [Online]. Available: https://www.chikamap.jp/chikamap/Portal?mid=216

[36] G. C. Pflug, "Some remarks on the value-at-risk and the conditional value-at-risk," in *Probabilistic Constrained Optimization*. Boston, MA, USA: Springer, 2000, pp. 272-281

[37] P. Krokhmal, S. Uryasev, and J. Palmquist, "Portfolio optimization with conditional value-at-risk objective and constraints," *J. Risk*, vol. 4, pp. 43–68, Mar. 2001.

APPENDIX A GENERAL MEAN-CVAR MODEL

Conditional value-at-risk (CVaR) was first proposed by Rockafellar and Uryasev [27] as a downside risk measure to quantify tail losses. In comparison with the traditional risk measure value-at-risk (VaR), CVaR has more attractive mathematical properties (e.g., subadditivity, convexity; see [36]). More importantly, CVaR can illustrate the losses exceeding VaR, but VaR does not control such losses.

Let $\mathcal{L}(\boldsymbol{x}, \boldsymbol{y})$ denote the loss function with respect to a decision vector $\boldsymbol{x} \in X \subset \mathbb{R}^n$ and uncertain vector $\boldsymbol{y} \in \mathbb{R}^m$ and $p(\boldsymbol{y})$ denotes the probability density function associated with $\boldsymbol{y}$. For a fixed decision vector $\boldsymbol{x}$, the cumulative distribution function of the loss can be written as:

$$\psi(\boldsymbol{x}, \alpha) = \int_{\mathcal{L}(\boldsymbol{x},\boldsymbol{y}) \leq \alpha} p(\boldsymbol{y})\, d\boldsymbol{y}. \tag{14}$$

Given a confidence level β, the β-VaR associated with the decision vector $\boldsymbol{x}$ is as follows:

$$\mathrm{VaR}_\beta(\boldsymbol{x}) = \min\{\alpha \in \mathbb{R} : \psi(\boldsymbol{x}, \alpha) \geq \beta\}. \tag{15}$$

Three values of β are commonly considered: 90%, 95%, and 99%. The β-CVaR associated with the decision vector $\boldsymbol{x}$ is defined as:

$$\mathrm{CVaR}_\beta(\boldsymbol{x}) = \frac{1}{1-\beta} \int_{\mathcal{L}(\boldsymbol{x},\boldsymbol{y}) \geq \mathrm{VaR}_\beta(\boldsymbol{x})} \mathcal{L}(\boldsymbol{x}, \boldsymbol{y}) p(\boldsymbol{y}) d\boldsymbol{y}, \tag{16}$$

which is the conditional expectation of the loss that is beyond VaR_β . It is difficult to deal with (16) directly, so Rockafellar and Uryasev [27] developed a simpler auxiliary function:

$$\mathcal{F}_\beta(\boldsymbol{x}, \alpha) = \alpha + \frac{1}{1-\beta} \int_{\boldsymbol{y} \in \mathbb{R}^m} [\mathcal{L}(\boldsymbol{x}, \boldsymbol{y}) - \alpha]_+ p(\boldsymbol{y}) d\boldsymbol{y} \tag{17}$$

where $[x]_+ = \max\{x, 0\}$ for $x \in \mathbb{R}$, and proved

$$\mathrm{CVaR}_\beta(\boldsymbol{x}) = \min_{\alpha \in \mathbb{R}} \mathcal{F}_\beta(\boldsymbol{x}, \alpha). \tag{18}$$

For practical applications, we often consider the following scenario-based approximation, i.e., (19), with a number of scenarios in the name of $\boldsymbol{y}^s$ for s in the scenario set S to avoid numerical difficulties caused by the integration in (17).

$$\mathcal{F}_\beta(\boldsymbol{x}, \alpha) \approx \alpha + \frac{1}{(1-\beta)|S|} \sum_{s \in S} [\mathcal{L}(\boldsymbol{x}, \boldsymbol{y}^s) - \alpha]_+ \tag{19}$$

The system operators usually try to minimize the risk while maximizing the return (or minimizing the loss). In this case, we can consider the mean-CVaR model, taking both return and risk into account. Krokhmal *et al*. [37] illustrated three equivalent formulations of the mean-CVaR model. A typical one among them is

$$\min_{\boldsymbol{x}, \alpha} \{\lambda \mathbb{E}[\mathcal{L}(\boldsymbol{x}, \boldsymbol{y})] + (1-\lambda) \mathcal{F}_\beta(\boldsymbol{x}, \alpha)\}, \tag{20}$$

where λ is the weight value, ranging from 0 to 1, and $\mathbb{E}[\cdot]$ is the expectation function. Considering discrete scenarios with the same probability, we can reformulate problem (20) in an approximate form:

$$\min_{\boldsymbol{x}, \alpha} \left\{ \frac{\lambda}{|S|} \sum_{s \in S} \mathcal{L}(\boldsymbol{x}, \boldsymbol{y}^s) + (1-\lambda) \left(\alpha + \frac{1}{(1-\beta)|S|} \sum_{s \in S} [\mathcal{L}(\boldsymbol{x}, \boldsymbol{y}^s) - \alpha]_+ \right) \right\}. \tag{21}$$

KAI ZHANG was born in Jingjiang, Jiangsu, China in 1994. He received the B.S. degree in transportation engineering from Yangzhou University, Yangzhou, China, in 2016, the M.S. degree in traffic information engineering & control from Nanjing University of Science & Technology, Nanjing, China, in 2019, and the M.S. degree in system management engineering from Fukuoka Institute of Technology, Fukuoka, Japan, in 2019. He is currently pursuing the Ph.D. degree in policy and planning sciences at University of Tsukuba, Ibaraki, Japan.

Since 2019, he has been a Research Assistant with University of Tsukuba. His research interest includes mathematical optimization and its application to transportation systems.

YUICHI TAKANO received his B.S. degree in policy and planning sciences in 2005, M.S. degree in engineering in 2007, and Ph.D. degree in engineering in 2010, all from the University of Tsukuba, Ibaraki, Japan.

From 2010 to 2014, he was an Assistant Professor with the Department of Industrial Engineering and Management, Tokyo Institute of Technology, Tokyo, Japan. In 2014, he joined the School of Network and Information, Senshu University, Kanagawa, Japan, as a Lecturer, and in 2016 became an Associate Professor. Since 2018, he has been an Associate Professor with the Faculty of Engineering, Information and Systems, University of Tsukuba. His primary research interests are mathematical optimization and its application to financial engineering and machine learning.

YUZHU WANG was born in Yichang, Hubei, China in 1994. He received the B.S. degree in mathematics and applied mathematics from Harbin Institute of Technology, Weihai, China, in 2016. He recived the M.S. degree in policy and planning sciences from University of Tsukuba, Ibaraki, Japan, in 2019. He is currently studying for the Ph.D. degree in policy and planning sciences at University of Tsukuba, Ibaraki, Japan.

He has been a Research Assistant at University of Tsukuba since 2019. He is currently interested in conic optimization problems, approximations of the semidefinite cone and algorithms for solving large-scale semidefinite optimization problems.

AKIKO YOSHISE was born in Machida, Tokyo, Japan in 1962. She received her B.S., M.S., and Ph.D. degrees, all in engineering, from Tokyo Institute of Technology in 1985, 1987, and 1990.

Since 2007, she has been a Professor with the Faculty of Engineering, Information and Systems, University of Tsukuba, Ibaraki, Japan, and is currently an Executive Officer of the university. She played a part in developing theoretical foundations of the primal-dual interior-point algorithm for linear programming and complementarity problems. Her research interests include mathematical optimization, especially conic optimization, and its applications in service science. She is an Editor of the journals *Numerical Algorithms*, *Japan Journal of Industrial and Applied Mathematics*, and *Pacific Journal of Optimization*.

Professor Yoshise was a recipient of the INFORMS Computing Society Prize in 1992 and the Frederick W. Lanchester Prize in 1993.