# Cooperative Routing Problem between Customers and Vehicles for On-demand Mobile Facility Services

Tomoki Nishi*, Keisuke Otaki, Ayano Okoso and Alex Fukunaga

***Abstract*— On-demand mobile facility services are a promising approach to mitigate social problems related to transportation. Route optimization to satisfy customer demands is an essential technology to realize the services. Most studies of the route optimization for the services have been focused on finding a better assignment from vehicles to customers and a better order of visiting customer locations under the assumption that the customers waiting at the locations without moving. In this paper, we formulate cooperative routing problem between customers and vehicles, which minimizes total travel cost by optimizing both vehicle and customer routes. We also propose a heuristic approach to find solutions for large instances. We demonstrate that customer cooperation helps to reduce the total travel cost compared to a solution of standard vehicle routing problem in synthetic experiments using the road network of Manhattan, NY, USA. We confirmed that the total travel cost of the customers and the vehicles was reduced by 20% using our heuristics comparing to solutions of the vehicle routing problem with little extra computational cost.**

## I. Introduction

*a) Background and motivation:* According to estimates by United Nations, worldwide, the population living in urban areas will increase to around 60% by 2050 [1]. This population concentration has a negative effect not only in urban areas but also in rural areas: aging of residents and depopulation. Mobility on demand services are a promising approach to mitigate the social issues related to transportation. Mobile facility services such as food trucks, mobile supermarkets and water trucks would help residents who have difficulties visiting restaurants and supermarkets due to reasons such as travel distance or health conditions.

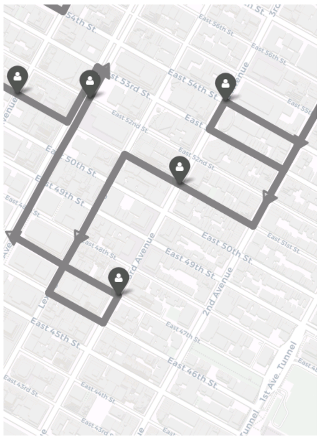
In such services, it would be ideal for vehicles to visit each customer location. However, this is often difficult due to several reasons. For example, the number of vehicles is insufficient or roads are too narrow for vehicles to pass. The situation would be mitigated with a little help from the customers in many cases. Figure 1(a) shows an example of the route in Manhattan, New York, USA, which found by solving vehicle routing problem (VRP) [2]. The vehicle needs to take detours to visit customers who are on one-way roads. On the other hand, if the customers travel a bit closer location to the vehicle route, the vehicle can save travel distance and time (Fig. 1(b)).

In this paper, we propose the CMFRP, a formulation for cooperative mobile facility routing problems among customers and vehicles. This formulation seeks to minimize the total travel cost by optimizing the routes of both vehicles and customers. This optimization would help not only the mobility service company but also the customers using the service because the number of customers that can use the service vehicles will increase by improving the travel distance and time of the vehicles.

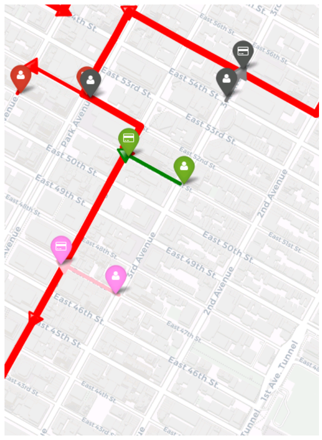
Tomoki Nishi, Keisuke Otaki and Ayano Okoso are with Toyota Central R&D Labs., Inc.,41-1 Yokomichi, Nagakute, Aichi, Japan

Alex Fukunaga is in the Department of General Systems Studies, Graduate School of Arts and Sciences, The University of Tokyo.

* Corresponding author, Email: nishi@mosk.tytlabs.co.jp



(a) a route with VRP



(b) a route with CMFRP

Fig. 1: Examples of routes: (a)solution of vehicle routing problem and (b) solution of our cooperative routing problem. Person icons denote customer locations and cash icons are the location where customers use a mobile facility (e.g., to take away foods).

*b) Statement of contributions:* We formulate CMFRP as an integer programming problem (IP) based on the cooperative routing problem among heterogeneous vehicles proposed by Otaki et al. [3]. We propose a heuristic approach for large instances of the CMFRP, which finds an initial solution by using a VRP solver and then locally updates a solution by iteratively solving subproblems. We experimentally evaluate our heuristic solver comparing to solutions of VRP using a road network of Manhattan, New York. We demonstrate that the total travel distance for the mobile facility services was reduced by 20% using our heuristics comparing to solutions of VRP with little extra computational cost.

## II. Related work

Mobile facility routing problem (MFRP) seeks to optimize routes for a fleet of mobile facilities when each location on the routes can provide service to several events [4]. The formulation maximizes the spatio-temporal distributed demand served by these mobile facilities during a continuous-time planning horizon, but does not consider costs for both vehicles and customers. Our CMFRP formulation finds routes to minimize costs for both vehicles and customers.

The multi-vehicle covering tour problem ($m$-CTP), which is similar to the MFRP, is the problem of finding vehicle routes that cover all demand points of customers, where the vehicles can cover several demand points at each location on the routes [5]. The $m$-CTP seeks minimal cost routes which cover all points, while the MFRP seeks routes that maximize the demand serviced by the facility. The $m$-CTP considers only vehicle travel costs, but not the customer costs of traveling (e.g., via walking) from a location to the event points served from the location. On the other hand, our CMFRP formulation considers non-vehicle travel costs incurred by the customers and minimizes the total travel costs for both vehicles and customers.

Otaki et al. proposed the cooperative routing problem for heterogeneous vehicle types [3], [6]. The problem models truck platooning and truck-drone cooperation for last-mile logistics. Their formulation seeks to find cooperative routes minimizing travel costs for vehicles, under giving table of effect for each cooperation among vehicle types (e.g., platooning between trucks reduces fuel consumption due to the air resistance reduction). We formulate the CMFRP by modifying their formulation. The difference is that in the formulation by Otaki et al., all type of vehicles have destinations, while the customers, which is a type of vehicles in their setting, have no destinations in the CMFRP because the goal of the customers is to use a mobile facility (e.g., to take away foods) but not to ride the mobile facility in order to go somewhere else.

The vehicle routing problem (VRP) and its many variations (e.g, capacitated VRP [7], VRP with time windows [8] and pickup-and-delivery problem [9]) have been widely studied, and many heuristics for finding solutions to large instances have been developed [2]. The VRP seeks to find vehicle routes to minimize the travel cost. In the VRP setting, vehicles travel to locations of customers and customers stay at initial location. The CMFRP is a generalization of the VRP, where both the vehicles and customers can move. Thus, in the heuristic we propose in Section IV, we use the solution returned by a VRP solver as an initial solution for the CMFRP.

## III. Cooperative Routing Problem between Customers and Vehicles

In this section, we propose a formulation of CMFRP, which is a problem to optimize paths of vehicles and customers to minimize a total travel cost. Table I summarizes the notation used in this paper. The formulation is developed on the model of cooperation among heterogeneous vehicles, proposed by Otaki et al. [3]. An instance of the CMFRP is defined by:

- a transportation network represented by a graph $G = \mathcal{V}, \mathcal{E})$ with nodes $\mathcal{V}$ and edges $\mathcal{E}$, where $\mathcal{V}$ represents locations and $\mathcal{E}$ represents roads between locations.
- a set of vehicles; the number of mobile facility vehicles is $N_k$. We use $[N_K]$ to denote $\{1, ..., N_k\}$, the set of indices corresponding to each vehicle.

TABLE I: Notation of CMFRP

| Symbol | Description |
|---|---|
| $N_C$ | the number of customers |
| $N_K$ | the number of vehicles |
| $w_{i,j,k}$ | A cost of vehicle $k$ to travel from $i$ to $j$ |
| $w_{i,j,c}$ | A cost of customer $c$ to travel from $i$ to $j$ |
| $G(\mathcal{V}, \mathcal{E})$ | A graph with a sef of nodes $\mathcal{V}$ and edges $\mathcal{E}$ |
| $Q_k$ | The maximum number of times customers can use vehicle $k$ |
| $o_k$ | an origin node of vehicle $k$ |
| $d_k$ | a destination node of vehicle $k$ |
| $o_c$ | an origin node of customer $c$ |
| $x_{i,j,k}$ | A binary variable where $x_{i,j,k} = 1$ when vehicle $k$ travel from $i$ to $j$ otherwise 0 |
| $x_{i,j,c}$ | A binary variable where $x_{i,j,c} = 1$ when customer $c$ travels from $i$ to $j$, or vehicle $v$ travels from $i$ to $j$ after customer $c$ uses the vehicle otherwise 0 |
| $\mu_{i,j}^{c,k}$ | A binary variable where $\mu_{i,j}^{c,k} = 1$ when customer $c$ already used vehicle $k$ before the vehicle travels from $i$ to $j$ otherwise 0 |
| $\mathbf{C}_k$ | A set of customers assigned to vehicle $k$ |
| $M_k$ | $[n_1, \ldots, n_{\lvert\mathbf{C}_k\rvert}]$; a list of the meeting nodes between customers and vehicle $k$ |

- a set of customers; the number of customers is $N_C$. We use $[N_C]$ to denote $\{1, ..., N_C\}$, the set of indices corresponding to each customer.
- vehicle/customer travel costs between locations in $G$, where the cost of a moving vehicle $k$ from $i \in \mathcal{V}$ to $j \in \mathcal{V}$ is denoted by $\mathbf{w}^{(K)} := \{w_{i,j,k}\}$, and the cost of a customer $c$ transporting himself/herself from $i \in \mathcal{V}$ to $j \in \mathcal{V}$ is denoted by $\mathbf{w}^{(C)} := \{w_{i,j,c}\}$.
- the initial locations of each vehicle $k \in [N_K]$, denoted by $\mathbf{o}^{(K)} := \{o_k\}$.
- the destinations of each vehicle vehicle $k \in [N_K]$, denoted by $\mathbf{d}^{(K)} := \{d_k\}$.
- the initial locations of each customer $c \in [N_C]$, denoted by $\mathbf{o}^{(C)} := \{o_c\}$.
- the maximum number of times customers can use each vehicle, denoted by $Q_k$, for each $v \in [N_K]$ .

Note that customers do not have specific destinations in this model because the goal of the customers is to use the services provided by a mobile facility vehicle (e.g., pick up food/groceries from the vehicle), and not to ride the vehicle to some destination.

We formulate CMFRP as an integer programming problem. The decision variables are the binary variables $\mathbf{x}^{(C)} := \{x_{i,j,c}\}$, $\mathbf{x}^{(K)} := \{x_{i,j,k}\}$ and $\boldsymbol{\mu} := \{\mu_{i,j}^{c,k}\}$, defined in Table I. $x_{i,j,c} = 1$ when customer $c$ travels from $i$ to $j$, or the vehicle used by customer $c$ travels from $i$ to $j$ after the customer uses the vehicle otherwise $x_{i,j,c} = 0$. $x_{i,j,k} = 1$ when vehicle $k$ travels from $i$ to $j$ otherwise $x_{i,j,k} = 0$. $\mu_{i,j}^{c,k} = 1$ when customer $c$ already used vehicle $v$ before the vehicle travels from $i$ to $j$ otherwise $\mu_{i,j}^{c,k} = 0$.

In addition, we use $\pi_c$ and $\pi_k$ to denote paths of customer $c$ and vehicle $k$, where a path is a sequence of nodes. The paths can easily be computed from the decision variables $\{x_{i,j,c}\}$, $\{x_{i,j,k}\}$ and $\{\mu_{i,j}^{c,k}\}$.

A solution to the CMFRP is a path for each vehicle and each customer. We seek a solution which minimizes the total travel cost incurred by all vehicles and customers for the following CMFRP;

$$\underset{\mathbf{x}^{(K)},\mathbf{x}^{(c)},\boldsymbol{\mu}^{c,k}}{\text{minimize}} \sum_{(i,j)\in\mathcal{E}} \sum_{k\in[N_K]} w_{i,j,k} x_{i,j,k} + \sum_{(i,j)\in\mathcal{E}} \sum_{k\in[N_K]} \sum_{c\in[N_C]} w_{i,j,c} \left( x_{i,j,c} - \mu_{i,j}^{c,k} \right), \quad \text{(1a)}$$

**subject to**

$$2\mu_{i,j}^{c,k} \leq x_{i,j,c} + x_{i,j,k}, \quad (\forall c \in [N_C], k \in [N_K], (i,j) \in \mathcal{E}), \quad \text{(1b)}$$

$$\sum_{k\in[N_K]} \mu_{i,j}^{c,k} \leq 1, \ (\forall c \in [N_C], (i,j) \in \mathcal{E}), \quad \text{(1c)}$$

$$\sum_{c\in[N_C]} \mu_{i,j}^{c,k} \leq Q_k, (\forall k \in [N_K], (i,j) \in \mathcal{E}), \quad \text{(1d)}$$

$$\sum_{j\in\mathcal{V}} x_{i,j,k} - \sum_{j\in\mathcal{V}} x_{j,i,k} = \begin{cases} 1 & \text{if } i = o_k, o_k \neq d_k \\ -1 & \text{if } i = d_k, o_k \neq d_k \\ 0 & \text{otherwise} \end{cases}, \quad (\forall i \in \mathcal{V}, k \in [N_K]), \quad \text{(1e)}$$

$$\sum_{j\in\mathcal{V}} x_{i,j,c} - \sum_{j\in\mathcal{V}} x_{j,i,c} = \begin{cases} 1 & \text{if } i = o_c, i \neq d_k, \\ 0 & \text{otherwise} \end{cases}, \quad (\forall i \in \mathcal{V}, k \in [N_K], c \in [N_C]), \quad \text{(1f)}$$

$$\sum_{j\in\mathcal{V}} x_{i,j,c} - \sum_{j\in\mathcal{V}} x_{j,i,c} \leq 0, \ (\forall i \in \{d_k\}, c \in [N_C]), \quad \text{(1g)}$$

$$\sum_{i\in\{d_k\}} \left( \sum_{j\in\mathcal{V}} x_{i,j,c} - \sum_{j\in\mathcal{V}} x_{j,i,c} \right) = -1, \ (\forall c \in [N_C]), \quad \text{(1h)}$$

$$x_{i,j,v}, x_{i,j,c}, \mu_{i,j}^{c,v} \in \{0,1\}. \quad \text{(1i)}$$

The first term of the objective function (1a) denotes the total travel cost of vehicles. The second term of the function is the cost of customers because we can obviously derive that $x_{i,j,c} - \mu_{i,j}^{c,k}$ is 1 on the edges $(i,j)$ customer $c$ travels and 0 after the customer used the vehicle $k$ (see the definitions of $x_{i,j,c}$ and $\mu_{i,j}^{c,k}$). The constraints (1b) indicate that $\mu_{i,j}^{c,k}$ can be 1 iff both $x_{i,j,c}$ and $x_{i,j,k}$ are 1. That is, $\mu_{i,j}^{c,k}$ can be 1 after customer $c$ uses vehicle $k$. The constraints (1c) represent that customer $c$ can use only one vehicle and the constraints (1d) denote that the number of customers that use vehicle $v$ is less than or equal to $Q_k$. The constraints (1e) and (1f) are flow conservation constraints for vehicles. The constraints (1g) and (1h), which are major difference from the formulation proposed by Otaki et al. [3], denote flow conservation constraints for customers and customer $c$ always uses only one of the vehicles respectively. The constraints are needed because the locations where the customers use the vehicles are not given in CMFRP in advance while origins and destinations are given in their formulation as requests. In our formulation, we can set different travel cost from $i$ to $j$ for each customer and each vehicle. For instance, some customers could travel less distance due to their health conditions than others.

CMFRP is obviously NP-hard problem because vehicle routing problem, which is a special case of CMFRP when customers stay at demand locations, is NP-hard. The number of decision variables are $|\mathcal{V}|^2 (N_C + N_K + N_C N_K)$ and of constraints are $|\mathcal{V}|^2 (N_C + N_K + N_C N_K) + |\mathcal{V}| N_C N_K + N_C$.

## IV. Proposed method

In this section, we propose our heuristic approach to find a solution of CMFRP based on two ideas: local update by solving subproblems and node reduction using distance constraints.

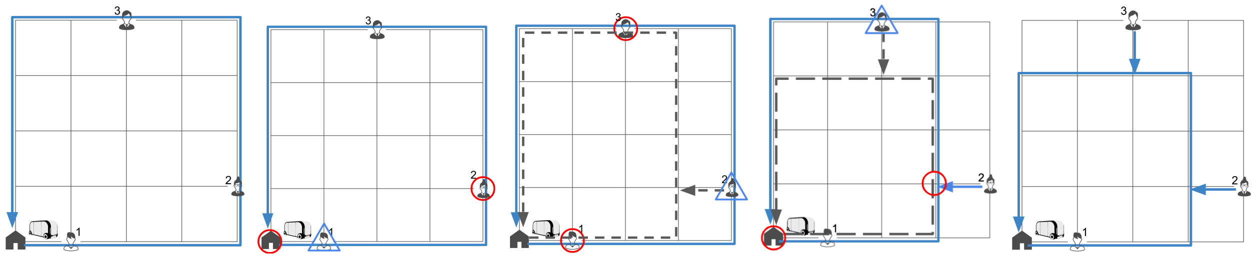
### A. Local update by solving subproblems

Our idea is to improve a solution iteratively applying a local update based on a feasible solution. Figure 2 (a)-(e) show examples of how the solutions are updated using our approach at each improvement iteration $t$. Icons of people, vehicle and house denote customers, vehicle and depot respectively. They can stay at grid points and travel adjacent grid points. The paths are represented by blue arrows in the figure. The vehicle leaves and returns to the depot while visiting the customers.

For example, in Fig. 2 (d), customers 1 and 3 wait at the initial locations without moving, and customer 2 travels one step at iteration $t = 3$. We can find a better solution (gray dashed arrows) by optimally solving a subproblem (reduced CMFRP instance) with only one customer (customer 3), and the vehicle departs from the meeting point and returns to the depot. The subproblem is more tractable than the original one because it only has one customer.

We show the update process in Algorithm 1. $\pi_k[0]$ and $\pi_c[0]$ are the first node in each path. $\pi_k[-1]$ is the last node in the path. SolveCMFRP$(G, \mathbf{w}, \mathbf{o}^{(K)}, \mathbf{d}^{(K)}, \mathbf{o}^{(C)})$ optimally solves the CMFRP using an IP solver such as Gurobi. $\mathbf{w}$ denotes $\{\mathbf{w}^{(K)}, \mathbf{w}^{(C)}\}$. SolutionToRoute$(\cdot)$ converts the solution returned by SolveCMFRP$(\cdot)$ to paths of vehicles and customers, and a set of vehicle assignment $\mathbf{C} := \{\mathbf{C}_k\}_{k\in[N_K]}$, where $\mathbf{C}_k$ denotes customers assigned to vehicle $k$. Let $\mathbf{M} := [M_1, \ldots, M_{N_K}]$ be a set of lists for meeting nodes between vehicle $k$ and customers in $\mathbf{C}_k$. Let $M_k := [n_1, \ldots, n_{|\mathbf{C}_k|}]$ be the list of the meeting nodes between customers and vehicle $k$. The meeting node for customer $c$ are defined as the first node where $x_{i,j,c} = 1$ and $\mu_{i,j}^{c,k} = 1$ on the path of customer $c$. $\boldsymbol{\pi}^{(K)} := \{\pi_k\}_{k\in[N_K]}$ is a set of vehicle paths and $\boldsymbol{\pi}^{(C)} := \{\boldsymbol{\pi}_{\mathbf{C}_k}\}_{k\in[N_K]}$ is a set customer paths, where $\boldsymbol{\pi}_{\mathbf{C}_k}$ denotes the path list for the customers assigned vehicle $k$. The list is sorted in order in which customers meet the vehicle. $N_{\text{update}}$ is the number of the updated vehicle paths.

The algorithm solves the subproblem defined by $G$, $\mathbf{w}$, $\mathbf{o}^{(K)}$, $\mathbf{d}^{(K)}$, and $\mathbf{o}^{(C)}$, which are computed from line 1 to line 3, to improve the current solution (i.e., $\boldsymbol{\pi}^{(K)}$ and $\boldsymbol{\pi}^{(C)}$) at line 4. We compute the updated solution and meeting points



(a) $t = 0$ (b) $t = 1$ (c) $t = 2$ (d) $t = 3$ (e) $t = 4$

Fig. 2: Example of update (allowing 1 customer to be moved at each improvement iteration $t$): Blue lines are current solutions and gray dashed lines are improved solution found by our local improvement method at each iteration. Red circles are a vehicle departure and goal locations. Blue triangles represent the customer allowed to move in current iteration.

from the decision variables returned by SolveCMFRP($\cdot$). We replace the current solution $\pi_k$ and $\boldsymbol{\pi}_{\mathbf{C}_k}$ to the updated solution $\hat{\pi}_k$ and $\hat{\boldsymbol{\pi}}_{\mathbf{C}_k}$ for vehicle $k$ and the customers assigned to the vehicle when $\hat{\pi}_k$ is not equal to $\pi_k$ (line 7-9).

When we consider only one customer in the subproblem, it constrains the update algorithm to preserve order of customers, i.e., Update($\cdot$) cannot find a solution which changes the order of customers served. Increasing the number of customers considered in the subproblem mitigates this issue and allows Update($\cdot$) to find solutions that swap some customers in the current order. That is, the better solution could be found by considering more customers in the subproblem, at the cost of incurring more computational effort to solve a larger subproblem.

**Algorithm 1** Update($G, \mathbf{w}, \boldsymbol{\pi}^{(K)}, \boldsymbol{\pi}^{(C)}$)

**Input:** $G, \mathbf{w}, \boldsymbol{\pi}^{(K)}, \boldsymbol{\pi}^{(C)}$
**Output:** $\boldsymbol{\pi}^{(V)}, \boldsymbol{\pi}^{(C)}, \mathbf{M}, \mathbf{C}, N_{\text{update}}$
1: $N_K, N_C \leftarrow |\boldsymbol{\pi}^{(K)}|, |\boldsymbol{\pi}^{(C)}|$
2: $\mathbf{o}^{(K)}, \mathbf{d}^{(K)} \leftarrow [\pi_k[0]]_{k\in[N_K]}, [\pi_k[-1]]_{k\in[N_K]}$
3: $\mathbf{o}^{(C)} \leftarrow [\pi_c[0]]_{c\in[N_C]}$
4: $\mathbf{x}^{(K)}, \mathbf{x}^{(C)}, \boldsymbol{\mu} \leftarrow$ SolveCMFRP($G, \mathbf{w}, \mathbf{o}^{(K)}, \mathbf{d}^{(K)}, \mathbf{o}^{(C)}$)
5: $\hat{\boldsymbol{\pi}}^{(K)}, \hat{\boldsymbol{\pi}}^{(C)}, \mathbf{M}, \mathbf{C} \leftarrow$ SolutionToRoute($\mathbf{x}^{(K)}, \mathbf{x}^{(C)}, \boldsymbol{\mu}$)
6: $N_{\text{update}} \leftarrow 0$
7: **for** $k$ in $\{k|\hat{\pi}_k \neq \pi_k\}$ **do**
8: $\quad \pi_k \leftarrow \hat{\pi}_k$
9: $\quad \boldsymbol{\pi}_{\mathbf{C}_k} \leftarrow \hat{\boldsymbol{\pi}}_{\mathbf{C}_k}$
10: $\quad N_{\text{update}} \leftarrow N_{\text{update}} + 1$
11: return $\boldsymbol{\pi}^{(V)}, \boldsymbol{\pi}^{(C)}, \mathbf{M}, \mathbf{C}, N_{\text{update}}$

### *B. Node reduction using distance constraints*

In many cases, we do not consider the entire input graph in Algorithm 1. We reduce the graph by eliminating nodes such that including them can only result in a solution which is dominated by the incumbent solution. Figure 3 illustrates this idea. The vehicle icon represents the vehicle that travels from customer 1 to customer 4. The blue arrows denote a current solution. All travel costs between pairs of adjacent nodes is 1. When we use Algorithm 1 to find a better solution (e.g., gray arrows), we do not need to consider the orange nodes because the cost of any vehicle paths visiting their nodes is always larger than the cost of the current path (i.e., $1 \rightarrow 2 \rightarrow 3 \rightarrow 4$), which is 5.

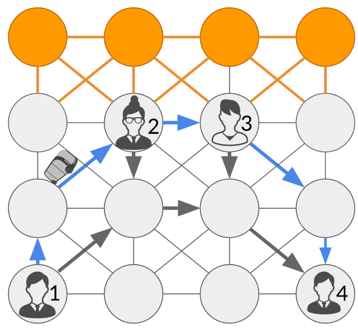



Fig. 3: Example of the nodes that can be reduced using distance constraints. Blue edges are the current paths and gray edges are the updated path. The orange nodes and edges do not need to be considered.

We use the induced graph with the nodes defined as follows;

$$\hat{\mathcal{V}} := \left\{ i \;\middle|\; \begin{array}{l} i \in \mathcal{V}, \\ w_{\pi_k} + w_{\boldsymbol{\pi}_c} \geq w_{o_k,i,k} + w_{i,d_k,k}, \\ o_k := \pi_k[0], d_k := \pi_k[-1], \forall k \in [N_K] \end{array} \right\}, \quad (2)$$

where $w_{u,v,k} := \min_{\pi_{u,v,k}} \sum_{(l,m)\in\pi_k} w_{l,m,k}$ and $w_{u,v,c} := \min_{\pi_{u,v,c}} \sum_{(l,m)\in\pi_c} w_{l,m,c}$ denote the lowest cost of the paths from $i$ to $j$ for each vehicle $k$ and customer $c$ respectively. $w_{\pi_k} := \sum_{(u,v)\in\pi_k} w_{u,v,k}$ is the path cost of vehicle $k$. Let $w_{\boldsymbol{\pi}_c} := \sum_{c\in\mathbf{C}_k} \sum_{(u,v)\in\pi_c} w_{u,v,c}$ be the total path cost of customers assigned to vehicle $k$. $\hat{\mathcal{V}}$ and $\hat{\mathcal{E}}$ are the nodes and edges of the reduced graph. The inequality in Eq. (2) indicates that the nodes with larger travel costs from an origin location than the current solution are reduced.

### *C. Proposed method*

We show our proposed method when simultaneously updating the paths of $N$ customers in Algorithm 2. We need an initial solution $\boldsymbol{\pi}^{(K)}$ and vehicle assignments $\mathbf{C}$ in order to apply Algorithm 1. We use the VRP solver with sophisticated heuristics to find an initial solution and vehicle assignment because CMFRP can recast to VRP when customers keep to stay at initial locations and the VRP solver is more efficient than IP solvers (line 1). We extract a subset of the current paths focused in next update at line 7, where $M_k[s]$ denotes $s$-th meeting node on the path of vehicle $v$ and $\pi_k[M_k[s] : M_k[e]]$ is a path from $M_k[s]$ to $M_k[e]$. Let

$\boldsymbol{\pi}_{\mathbf{C}_k}[s:e]$ represents the paths from $s$-th customer to $e$-th one on the vehicle path. The induced subgraph is constructed using Eq. (2) from line 9 to line 10. We update the paths using Algorithm 1 at line 11 and 12. The process is continued while improving the paths.

**Algorithm 2** SolveCMFRP$_{\text{heu}}(G, \mathbf{w}, \mathbf{o}^{(K)}, \mathbf{d}^{(K)}, \mathbf{o}^{(C)})$

**Input:** $G, \mathbf{w}, \mathbf{o}^{(K)}, \mathbf{d}^{(K)}, \mathbf{o}^{(C)}$
**Output:** $\boldsymbol{\pi}^{(K)}, \boldsymbol{\pi}^{(C)}, \mathbf{C}$
1: $\boldsymbol{\pi}^{(K)}, \mathbf{M}, \mathbf{C} \leftarrow$ SolveVRP$(G, \mathbf{w}, \mathbf{o}^{(K)}, \mathbf{d}^{(K)}, \mathbf{o}^{(C)})$
2: **for** $k$ in $[N_K]$ **do**
3: $\quad N_{\text{update}} \leftarrow 1$
4: $\quad$ **while** $N_{\text{update}} > 0$ **do**
5: $\quad\quad N_{\text{update}} \leftarrow 0$
6: $\quad\quad$ **for** $s \leftarrow 1, e \leftarrow N; e \leq |\mathbf{C}_k| - 1; s{+}{+}, e{+}{+}$ **do**
7: $\quad\quad\quad \hat{\boldsymbol{\pi}}^{(K)} \leftarrow \{\pi_k[M_k[s-1] : M_k[e+1]]\}$
8: $\quad\quad\quad \hat{\boldsymbol{\pi}}^{(C)} \leftarrow \{\boldsymbol{\pi}_{\mathbf{C}_k}[s:e]\}$
9: $\quad\quad\quad \hat{\mathcal{V}} \leftarrow$ Computed using Eq. (2)
10: $\quad\quad\quad \hat{G} \leftarrow$ Subgraph$(G, \hat{\mathcal{V}})$
11: $\quad\quad\quad \hat{\boldsymbol{\pi}}^{(K)}, \hat{\boldsymbol{\pi}}^{(C)}, \hat{\mathbf{M}}, \hat{\mathbf{C}}_k, \text{cnt} \leftarrow$ Update$(\hat{G}, \mathbf{w}, \hat{\boldsymbol{\pi}}^{(K)}, \hat{\boldsymbol{\pi}}^{(C)})$
12: $\quad\quad\quad \pi_k, \{\pi_c\}_{c \in \mathbf{C}_k}, \mathbf{C}_k, M_k \leftarrow \hat{\boldsymbol{\pi}}^{(K)}, \hat{\boldsymbol{\pi}}^{(C)}, \hat{\mathbf{C}}_k, \hat{\mathbf{M}}[0]$
13: $\quad\quad\quad N_{\text{update}} \leftarrow N_{\text{update}} + \text{cnt}$
14: return $\boldsymbol{\pi}^{(K)}, \boldsymbol{\pi}^{(C)}, \mathbf{C}$

## V. Experiments

### A. Experiment Setup

We evaluated our formulation and heuristics using a road network of Manhattan, New York, USA extracted from OpenStreetMap [10]. The number of nodes and directed edges are 4573 and 9871. We set the number of vehicles $N_K = 1$. We evaluated the total travel cost and the runtime when the number of customers $N_C$ are 50, 100, 200, and 500. The initial locations of the vehicles $\{o_k\}$ and the customers $\{o_c\}$ are selected uniformly at random from the set of nodes. We set the destinations of the vehicle $\{d_k\}$ to the same nodes as their initial locations. The travel costs between each pair of adjacent nodes $(i, j)$ for the vehicle $\{w_{i,j,k}\}$, and the customers $\{w_{i,j,c}\}$ were set to the travel distance between $i$ and $j$. We evaluated our heuristics at only $N_k = 1$ because our heuristics do not change the assignment itself due to update customer paths on an assigned vehicle path.

In the experiments, we compared CMFRP to VRP for the travel cost and runtime. We also evaluated the difference among the number of customers updated simultaneously using Algorithm 1.

In order to calculate Eq. (2) in Algorithm 2, we need to evaluate the minimal cost of traveling from $o_k$ to $i \in \mathcal{V}$ and from $i \in \mathcal{V}$ to $d_k$ $N_K$ times. In practice, this dose not incur significant cost, as we use the Hub-Labeling method [11], which can return minimal path costs among any pair in continental size networks in less than 1 millisecond, after an initial preprocessing overhead (14 minutes for continental size networks in their article, 76 seconds in our experiments with our naive implementation).

We evaluated the performance on a PC with an Intel Xeon CPU Broadwell@2.6 GHz and 112 GB memory. We solved the IP formulation of CMFRP by calling an LP solver included in Gurobi-8.1.1 [12] from a Python code. We solved the VRP using Google OR-Tools [13].

### B. Results

First, we tested to solve the instances without our heuristics using Gurobi. We, however, found that no feasible solutions were returned in 10 hours even for relaxed IP instances with 50 customers. Therefore we only show the results of our heuristics for CMFRP in this section.

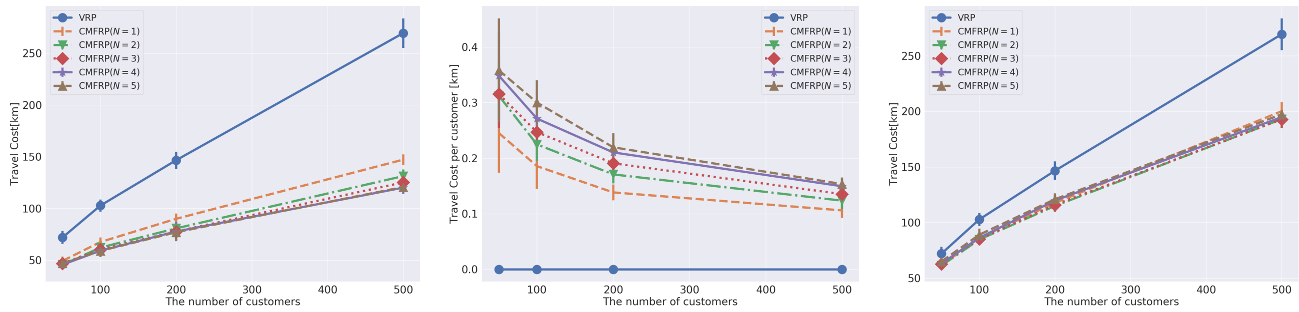
*1) Travel cost:* We evaluated travel cost with respect to the number of customers. Figure 4 shows the vehicle travel cost, the travel cost per customer and total travel cost for each approach. The blue line denotes the performance of paths optimized by VRP and other lines represent the ones by CMFRP. Figure 4 (a) indicates that the paths optimized by CMFRP significantly reduce vehicle travel cost comparing to the VRP solutions. Figure 4 (b) also shows that the travel cost per customer was several hundreds meters, and the effect of increasing the travel cost on each customer was limited. The travel cost of customers for VRP is of course zero because the customers keep to stay at their initial locations. The total travel cost (including vehicles and customers) was reduced by 20%, compared to the VRP formulation from Fig. 4 (c). Note that as the number (and therefore the density) of customers increase, the distance from their initial locations to the locations where they are served by the vehicle tends to decrease, resulting in lower customer travel costs.

We also evaluate the effect of varying the number of customers $N$, which is updated simultaneously in Algorithm 1. The results showed in Fig.4 indicated that although $N$ had an effect on both vehicle and customer travel costs considered separately, there was little effect on the total travel cost.

*2) Runtime:* We evaluated the runtime of the methods with respect to the number of customers. Figure 5 shows the runtime of each approach: (a) the heuristics without search node reduction presented in subsection IV-B (i.e., Algorithm 2 without line 8 and 9) and (b) the heuristics with the node reduction (Algorithm 2). The blue line denotes the runtime of the VRP solver and the other lines represent the one of our heuristics with varying number of customers updated simultaneously in the local updates. The results indicate that the heuristics with the node reduction can find solution with much lower computational cost than without the node reduction. The results in Fig. 5 (b) show that the runtimes increase as $N$, the number of customers included in the subproblems solved by Update. However, as discussed above, Fig. 4 shows that $N$ has little effect on total travel cost, so in practice, the tradeoff between the additional runtime due to increasing $N$ and the reduction in travel costs needs to be carefully considered.

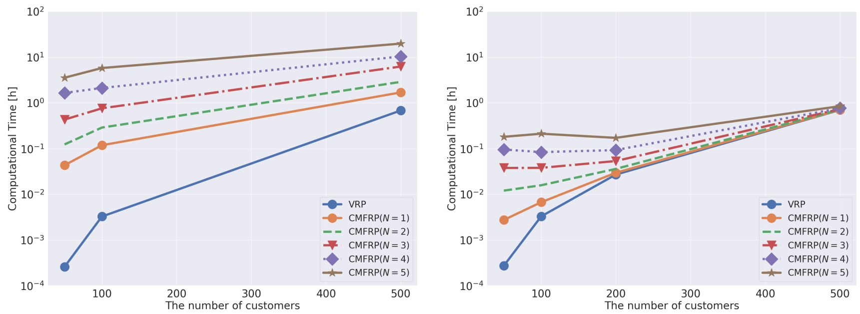
## VI. Concluding Remark

In this paper, we proposed the cooperative mobile facility routing problems among customers and vehicles, named CMFRP. We formulated CMFRP as an integer programming problem. We also developed a heuristic approach to finding feasible solutions in larger instances efficiently. We presented a local update by solving a lot of small CMFRP instances iteratively after finding initial solutions using a VRP solver.




(a) Vehicle travel cost (b) travel cost per customer (c) Total travel cost

Fig. 4: Travel cost optimized using VRP and CMFRP against the number of customers: (a) vehicle travel cost, (b) travel cost per customer and (c) total travel cost. Blue lines denotes the results of paths optimized by VRP. The other lines are the results solving CMFRP. $N$ denotes the number of customers updated simultaneously using Algorithm 1.




(a) Heuristics without search node reduction (b) Heuristics with search node reduction

Fig. 5: Runtime for each method with respect to the number of customers. (a) heuristics without search node reduction and (b) heuristics with search node reduction. Blue lines denotes the results of paths optimized by VRP. The other lines are the results solving CMFRP. $N$ denotes the number of customers updated simultaneously using Algorithm 1.

We also developed the method to reduce the number of nodes in the small sub-problems using distance constraints.

We evaluated our heuristics with numerical experiments using the road network of Manhattan, New York, USA extracted from OpenStreetMap. The results showed that the paths found by CMFRP reduced 20% of the total travel cost comparing to paths by VRP. We also confirmed that the effect of the number of customers updated simultaneously in Algorithm 1 is little against the travel cost. Therefore a small number of customers would be enough to reduce the total travel cost in practice. Moreover, we also confirmed that the travel cost of each customer was limited. We analyzed the increase in the runtime of our heuristics comparing to the VRP solver. Our search node reduction helped to reduce the runtime in our heuristics. Moreover, our heuristics needed little extra runtime to find cooperative paths among customers and vehicles. Our approach cannot change the assignment itself because updating paths of customer on a assigned vehicle path. We will develop such heuristics updating vehicle assignments as future work.

## REFERENCES

[1] UN, "World urbanization prospects: The 2014 revision population database." Tech. Rep., 2014.

[2] D. Pisinger and S. Ropke, "A general heuristic for vehicle routing problems," *Computers & Operations Research*, vol. 34, no. 8, pp. 2403–2435, aug 2007.

[3] K. Otaki, S. Koide, A. Okoso, and T. Nishi, "Coopera-tive Routing with Heterogeneous Vehicles," in *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*, 2019, pp. 2150—-2152.

[4] R. Halper and S. Raghavan, "The Mobile Facility Routing Problem," *Transportation Science*, vol. 45, no. 3, pp. 413–434, aug 2011.

[5] M. H. Hà, N. Bostel, A. Langevin, and L.-M. Rousseau, "An exact algorithm and a metaheuristic for the multi-vehicle covering tour problem with a constraint on the number of vertices," *European Journal of Operational Research*, vol. 226, no. 2, pp. 211–220, 2013.

[6] K. Otaki, S. Koide, A. Okoso, and T. Nishi, "Multi-Agent Path Planning with Heterogeneous Cooperation," in *Proceedings of the IEEE International Conference on Tools with Artificial Intelligence*, 2019.

[7] G. Laporte and F. Semet, "Classical heuristics for the capacitated vrp," in *The vehicle routing problem*. SIAM, 2002, pp. 109–128.

[8] J.-F. Cordeau and Q. Groupe d'études et de recherche en analyse des décisions (Montréal, *The VRP with time windows*. Groupe d'études et de recherche en analyse des décisions Montréal, 2000.

[9] G. Desaulniers and Q. Groupe d'études et de recherche en analyse des décisions (Montréal, *The VRP with pickup and delivery*. Groupe d'études et de recherche en analyse des décisions, 2000.

[10] M. Haklay and P. Weber, "Openstreetmap: User-generated street maps," *IEEE Pervasive Computing*, vol. 7, no. 4, pp. 12–18, 2008.

[11] I. Abraham, D. Delling, A. V. Goldberg, and R. F. Werneck, "A hub-based labeling algorithm for shortest paths in road networks," in *International Symposium on Experimental Algorithms*. Springer, 2011, pp. 230–241.

[12] Gurobi Optimization, LLC, "Gurobi optimizer reference manual," 2018. [Online]. Available: http://www.gurobi.com

[13] L. Perron and V. Furnon, "Or-tools," Google. [Online]. Available: https://developers.google.com/optimization/