City-wide Smart Package Distribution Using Crowdsourced Public Transportation Systems

Geyao Cheng, Deke Guo, Senior Member, IEEE, Jianmai Shi, and Yudong Qin

Abstract—The demand for package delivery services is extremely huge every day, because of the rapid development on online retailers. This leads to huge traffic congestion, resource consumption and environmental pollution (e.g. carbon emission). However, the urban public transport system offers a large amount of under-utilized capacity outside the peak hours. In this paper, we present the City-wide Package Distribution problem using Crowdsourced Public Transportation Systems (CPTS). That is, packages are delivered by carefully utilizing the considerable amount of idle capacity of the CPTS. Specifically, given a number of packages and the timetable of available CPTS trips, we optimize the package delivering scheme by determining the four identified states of any package at any time slot (waiting, riding, re-waiting and being unloaded). The delivering scheme can be modeled as an instance of the multi-commodity flow problem, and formulated by the Mixed Integer Linear Programming techniques. We then propose an efficient heuristic solution for this NP-hard problem. Finally, our work is validated via comprehensive simulations with a real bus transportation network.

Index Terms—package distribution, crowdsourced public transportation system, idle capacity, package states, MILP.

I. INTRODUCTION

The rapid development on online retailers leads to the huge demand of package delivery services, which urges the sustainable development of logistics industry [1] [2]. Specifically, Chinese postal enterprises have built more than 1,000 warehousing and distribution centers, and newly opened 153 trunk postal routes in 2017 [3]. Although the logistics grow significantly, these companies still face many challenges in the package distribution problem. One of the main challenges is providing convenient same-day delivery services [4] in an efficient way. However, the large number of packages to be delivered gives rise to the continuous increase of the consumed manpower and other resources [5]. Data shows that, online ordered products have generated over one billion package distributions in 2013, and this number is predicted to grow by 28.8% in 2018 [6]. Furthermore, the dedicated urban vehicles for logistics would incur serious air pollution and traffic congestion, especially when the traffic volume is huge [7]. Therefore, cities are looking for instruments and policies to ensure an efficient and effective urban transmission for both passengers and packages.

The package delivery problem has a trend to be solved by utilizing different transportation tools, such as unmanned aerial vehicles (UAV) [8], taxis [7], and private cars [9]. These researches aim at reducing the resource consumption and releasing traffic congestion. The first kind of efforts focuses on the mixed logistics [2] [10] [11] [12], where different kinds of vehicles would be successively utilized to realize the package deliveries, such as trucks & city freighters [13]. The second kind of efforts is based on Crowdsourced Delivery, where the package delivery tasks are outsourced in a free and voluntary manner [9] [14].

Although many kinds of researches have been made to realize the efficient and effective urban transmission, we urgently need a way to solve the package distribution problem from a global perspective and achieve the overall advantage. Public Transportation Systems (PTS) [15] would be competent to accomplish the package delivery tasks. As a urban infrastructure, the PTS is stable, time-scheduled, economically friendly, and widely covered [15]. Moreover, it exhibits a huge amount of under-utilized capacity at off-peak hours, which gives probability to the capacity share between passengers and packages. Inspired by such observations, we propose a novel idea of realizing the City-wide Package Distribution using Crowdsourced Public Transportation Systems, named the CPDCP problem. It is the first attempt to deliver packages to their final destinations economically and ecologically, utilizing the under-utilized capacity of Crowdsourced Public Transportation Systems (CPTS) vehicles. Thus, passengers and packages would share the capacity of CPTS vehicles together, and the capacity utilization of the CPTS could be significantly enhanced.

The business model of CPDCP is really feasible in the future, due to its abundant idle capacity, effectiveness, and environmental friendliness. The stops of CPTS have pervaded our cities with a high coverage rate. This facilitates package deliveries in the first/last kilometer. Moreover, CPDCP cause...
less extra resource consumption and environmental pollution, because of the share of the idle capacity in CPTS vehicles between passengers and packages. Actually, some other transportation systems have already been adopted as part of new logistic methods in the commercial sector called last-mile delivery, such as drones [8] and taxis [7].

The process of CPDCP problem usually consists of three stages. In the first stage, each package is allocated to a bus route which has the shortest sum distance to the package’s original location and final destination. For city-wide packages, couriers pick up the packages at each street/block, and then transmit the packages to nearby bus stops. For country-wide packages, the logistic companies sort them at Consolidation and Departure Center (CDC), and deliver them to the departure stop of the dedicated bus route. In the second stage, packages wait at their start stops and then ride a trip to their target stops. Specifically, when under-utilized trips come, the packages would be loaded in a determined sequence and finally be unloaded when arriving at their target stops. Note that in the delivering process, the packages could be unloaded at the intermediate bus stops so as to vacate capacity for ensuring passenger experience in the busy time. In the third stage, the local freighters would collect the unloaded packages and distribute them to their final destinations.

In this paper, we mainly focus on solving issues involved in the second stage. Specifically, given a set of packages to be delivered, we plan a scheme using the minimum number of continuous under-utilized trips such that their idle capacities are sufficient for delivering such packages. Any package may have four states in the delivering process. As Fig. 1 shows, a package, whose start stop is $P_1$ and target stop is $P_4$, would wait at its start stop $P_1$ for an under-utilized trip, and then ride a bus from $P_1$ to $P_2$. When the package is unloaded at the intermediate stop ($P_2$ and $P_3$), its state becomes re-wait until the following under-utilized trip comes; finally, the state of the package would remain as unloaded after arriving at $P_4$. We derive a mathematical model for packages being delivering to their final destinations by determining their states at any time slot. Our model considers the temporal/spatial availability of the amount of idle capacity in the given bus trips in the CPTS. Thus, the experience of passengers would not be impacted. We model the scheduling task of a given set of packages as an instance of the multi-commodity flow problem, and formulated it using Mixed Integer Linear Programming techniques (MILP). Thereafter, we propose an efficient heuristic solution for this NP-hard problem.

We conduct extensive evaluations to verify the effectiveness of our solution for the CPDCP problem. Results show that more than 60% packages could be delivered to their target stops within 500m to their final destinations, when the CPTS involves 10 routes. Considering 1000 packages to be delivered, our heuristic method only occupies 7.5 trips in their off-peak working hours, and just consumes less than 4 hours to accomplish the delivering task.

The rest of this paper is organized as follows. Section II introduces the related work. Section III demonstrates the framework of CPDCP. In Section IV, we discuss the system model, the MILP-formulation, and the heuristic solution. Section V reports our experimental results. Finally, Section VI concludes this paper.

II. RELATED WORK

There is an increasing attention of researches aim to reduce the unnecessary resources consumption and environmental pollution caused by the dedicated transmission of packages.

Mixed logistics. The first kind of efforts focuses on mixed logistics, where more than one kind of vehicles would be successively utilized to realize the package deliveries. For example, Literature [12] introduced a heterogeneous delivery team of two cooperating vehicles: a truck carries a shipment of packages to the street blocks, and then city freighters carry individual packages from the blocks to the specific delivery points in the region. There are still other kinds of vehicles used in mixed logistics, such as buses and city freighters [2] [10], close-open mixed two-echelon [11], and trucks and Micro Aerial Vehicles [13].

Crowdsourced delivery. The second kind of efforts focuses on Crowdsourced Delivery, where the tasks of package deliveries are outsourced to a non-specific mass network in a free and voluntary manner. Some efforts aimed at having packages take hitchhiking rides, such as taxis [7] and private cars [9]. For example, Chen et al. [7] suggested that voluntary taxi drivers would pick up the passengers in cooperating with collecting packages. There were some other efforts optimizing the assignment of package delivery tasks to drivers. Setzke et al. [14] optimized the assignment of packages to drivers, subject to transportation routes and time constraints. Arslan et al. [9] proposed a peer-to-peer platform to realize the matching between packages and drivers. Additionally, drivers were suggested to accept several requests in one trip [16] for less time and resource consumption, compared to those direct deliveries accepting one request in one trip.

There is an trend to utilize different kinds of transportation tools, such as unmanned aerial vehicles (UAV) [8] and taxies [7], to realize the package deliveries. These researches aim at reducing the resource consumption and releasing traffic congestion. However, it remains open to deliver packages utilizing the under-utilized capacity of the Crowdsourced public transportation system. A few proposes are literature [2] and [10], which mainly focused on the third stage in our proposed CPDCP problem in this paper. Masson et al. [2] optimized the handshaking process between the buses and the city freighters. The primary objective of literature [10] was to minimize the number of city freighters used and the total time traveled by these vehicles. In this paper, we mainly focus on solving issues involved in the second stage. Specifically, given a set of packages to be delivered, we plan a scheme using the minimum number of continuous under-utilized trips such that their idle capacities are sufficient for delivering such packages.

III. FRAMEWORK OF CPDCP

In this section, we start with definitions of some basic concepts, and then formally state the framework of CPDCP.

A. Basic definitions

We first propose several important definitions which would be used in the remainder of this paper.

JOURNAL OF ...
Our proposed CPDCP model. (b) Our proposed CPDCP model.

Definition 1 (Route/Trip): We define a route as the travel sequence of a bus, linked by all involved bus stops alongside. A trip is a specific route starting at a scheduled time [17].

Definition 2 (Original location/Final destination): We define the place where a package is generated as Original location, and the place where a package aims at as Final destination.

Definition 3 (Start stop/Target stop): With a selected route, the bus stop where the package is loaded initially is defined as Start stop, and the bus stop which the package is unloaded finally is defined as Target stop. Note that, Start stop/Target stop is selected as the one having the shortest distance to the package’s Original location/Final destination, separately.

B. Framework of CPDCP

In the conventional logistic distribution model, as Fig. 2(a) shows, each logistic company distributes packages from their subordinate CDC to their customers independently and separately, using their own vehicles (both trucks and local freighters). Therefore, the large number of operating vehicles would incur more resource input, traffic congestion, and air pollution. However, our proposed CPDCP, as shown in Fig. 2(b) solve such problems greatly.

The main idea of CPDCP is delivering packages from their Original locations to their Final destinations economically and ecologically, utilizing the considerable idle capacity of CPTS vehicles (from Start stops to Target stops). The process usually consists of three stages. In the first stage, packages are delivered to the Start stops of their selected bus routes. Taking city-wide package delivery as an example, couriers pick up the packages at each street/block, and then transmit each package to the nearby bus stops of a bus route which has the shortest distance to the package’s Original location and Final destinations. Obviously, the selected CPTS routes must have cooperation with the logistics companies. In the second stage, packages are delivered from their Start stops to their Target stops, riding the selected routes. Note that, in the delivering process, the packages would be unloaded at the intermediate stops at a busy time, for ensuring the experience of passengers. In the third stage, the local freighters would collect the unloaded packages and distribute them to their Final destinations.

Compared to the existed logistics distribution model that maintains dedicated shipping crews, our CPDCP model has the following advantages. 1) Effectiveness. The bus stops are usually close to the densely populated areas. The Flint Hills Area Transportation Agency report states that 75% of off-campus students and 35% of employees live within five minutes of the bus city-wide routes [18]. Moreover, the stops of CPTS have pervaded our cities with a high coverage rate [19]. This facilitates the delivery tasks in the first/last kilometer [20]. 2) Considerable amount of under-utilized capacity. Bus exhibits a huge amount of under-utilized capacity in off-peak hours, which offers a great chance for sharing capacity between passengers and packages. Moreover, the CPTS operates stably with a time schedule [21]. Thus, the packages would be delivered to their target destinations in the operational time, with high feasibility. 3) Economic and environmental friendliness. Our proposed solution only leverages the under-utilized capacity of the CPTS vehicles for package distribution. Thus, it induces little extra air and noise pollution [5]. Moreover, we cluster packages from the CDCs of different logistic companies and then deliver them to their Final destinations from a global perspective. This would reduce the total travel distance, resulting in less resource consumption, traffic consumption, and air pollution.

Additionally, the CPDCP incurs a low impact on the quality of passenger experience. We only utilize the under-utilized capacity of the CPTS vehicles for logistics. Thus, the seats for passengers would not be occupied. In addition, the original stopping time at each bus stop facilitates loading and unloading packages. Therefore, package distribution would cause less additional waiting time for passengers. Additionally, couriers, selected bus routes, selected bus trips, and the freighters all have an ID. Thus, packages are traceable in the whole delivery process, which guarantees the package security and avoids delivery mistakes or losses.

IV. CPDCP Problem Solution

In this section, we first define the system model and its main objectives (Section IV-A). Then we address the problem via MILP technique (Section IV-B, and provide an efficient heuristic solution (Section IV-C). Finally, we expand the model from one single route to the whole traffic network (Section IV-D).

A. Problem Statement

Given a route, the schedule at bus stop \( s \) is a nearly deterministic arrival (and departure) process. We consider a time slotted model \( \xi_s = \{t^s_k : k=1,2,\ldots\} \), where \( t^s_k \) represents the time slot when the \( k^{th} \) trip arrives at bus stop \( s \). The length of each slot can be determined by historical data about bus arrival and departure. The scenario can be defined as a tuple \( \langle S,P,TP \rangle \), where:

- \( S = \{s_1,s_2,\ldots,s_{|S|}\} \) is the set of bus stops of the given Crowdsourced bus route.
- \( P = \{p_{1,2,\ldots,P}\} \) represents the set of packages to be delivered using the given route. Specifically, the information of each package can be presented as a tuple \( p_l = p_{l,o,d}\{s,k,x,y,z,v\} \). Let \( o \) and \( d \) represent the corresponding Start stop and Target stop of the package, separately. Let \( s \) represent the current located bus stop and \( k \) represent the upcoming trip of that route. Additionally,
### TABLE I
**Notations of model formulation for a given route**

<table>
<thead>
<tr>
<th>Indices:</th>
<th>Sets:</th>
<th>Parameters:</th>
<th>Decision variables:</th>
</tr>
</thead>
<tbody>
<tr>
<td>$s/d$</td>
<td>$S$</td>
<td>$Q_i^k$</td>
<td>$x_{p_i,s,k}$</td>
</tr>
<tr>
<td>$k$</td>
<td>$P$</td>
<td>$v_{p_i,k}$</td>
<td>$y_{p_i,s,k}$</td>
</tr>
<tr>
<td>$i$</td>
<td>$TP$</td>
<td>$\text{vol}_{p_i}$</td>
<td>$z_{p_i,s,k}$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>$v_{p_i,s,k}$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>$w_{s,k}$</td>
</tr>
</tbody>
</table>

$x, y, z, v$ represents four states of the delivering package, i.e., waiting, riding, re-waiting, and being unloaded, separately.

- $TP = \{k_1, k_2, \ldots, k_{|TP|}\}$ represents the set of trips of the given bus route, which operates continuously in a time schedule.

Based on the previous definition, we assume that any package can be delivered to their Final destinations using a bus route in the CPTS. In the second stage of our proposed CPDCP problem, for a given route, when the $k$th trip arrives at its $s^{th}$ bus stop, i.e., at time slot $t_s^k$, each package $p_i$ is in one of the following four states:

- waiting, i.e., staying at the **Start stop**. We introduce the following binary state variable $x_{p_i,s,k} \in \{0, 1\}$ for any $p_i \in P$, $t_s^k \in \xi$, such that:
  \[
  x_{p_i,s,k} = \begin{cases} 
  1 & \text{if } p_i \text{ stays at its start stop at time } t_s^k \\
  0 & \text{otherwise}.
  \end{cases}
  \]  

- riding, i.e., riding a bus and aiming at its **Target stop**. We introduce the following binary state variable $y_{p_i,s,k} \in \{0, 1\}$ for any $p_i \in P$, $t_s^k \in \xi$, such that:
  \[
  y_{p_i,s,k} = \begin{cases} 
  1 & \text{if package } p_i \text{ rides a bus at time } t_s^k \\
  0 & \text{otherwise}.
  \end{cases}
  \]  

- re-waiting, i.e., being unloaded at the intermediate bus stop because of the capacity limitation. We introduce the following binary state variable $z_{p_i,s,k} \in \{0, 1\}$ for any $p_i \in P$, $t_s^k \in \xi$, such that:
  \[
  z_{p_i,s,k} = \begin{cases} 
  1 & \text{if } p_i \text{ stays at an intermediate bus stop at time } t_s^k \\
  0 & \text{otherwise}.
  \end{cases}
  \]  

- being unloaded, i.e., having being unloaded at its **Target stop**. We introduce the following binary state variable $v_{p_i,s,k} \in \{0, 1\}$ for any $p_i \in P$, $t_s^k \in \xi$, such that:
  \[
  v_{p_i,s,k} = \begin{cases} 
  1 & \text{if } p_i \text{ has arrived at its target stop at time } t_s^k \\
  0 & \text{otherwise}.
  \end{cases}
  \]

Let $T(p_i, t_s^k)$ refer to the consumed time of delivering package $p_i$ at time $t_s^k$. Based on the state variables defined above, for any package $p_i$, $T(p_i, t_s^k)$ is updated at each $t_s^k \in \xi$ as follows:

\[
T(p_i, t_s^k) = (x_{p_i,s,k-1} + z_{p_i,s,k-1}) \cdot [T(p_i, t_s^{k-1}) + t_s^k - t_s^{k-1}] \\
+ y_{p_i,s-1,k} \cdot [T(p_i, t_s^k-1) + t_s^k - t_s^{k-1}] \\
+ v_{p_i,s-1,k} \cdot T(p_i, t_s^{k-1}) + v_{p_i,s,k-1} \cdot T(p_i, t_s^{k-1})
\]  

Based on the value of $T(p_i, t_s^k)$, we introduce an additional state-transfer-off decision variable $w_{s,k} \in \{0, 1\}$ for any $p_i \in P$ at time $t_s^k$ such that:

\[
w_{s,k} = \begin{cases} 
  1 & \text{if } v_{p_i,s-1,k} = 1 \text{ for any } p_i \in P \text{ at time } t_s^k \\
  0 & \text{otherwise}.
  \end{cases}
  \]

The delivering time $DT$ is defined as the minimum time slot such that $DT > w_{s,k} \times t_s^k$. Informally, the CPDCP problem can be defined as: determine the variable statues $x_{p_i,s,k}, y_{p_i,s,k}, z_{p_i,s,k},$ and $v_{p_i,s,k}$ for any package $p_i$ and any time slot $t_s^k$, so that $DT$ is minimized. The constraint is that all packages are delivered to their Target stops by the CPTS. In the following, we prove that this model is a special instance of the multi-commodity flow model, which can be formulated via MILP techniques.

**B. MILP-based formulation**

We devise a multi-period directed multi-graph $G(V, A)$ in order to model the movement of packages across the **Start stop**, **Target stop**, and the intermediate stops. For a given bus route, the set of nodes $V$ is composed by the couple $(s, k)$, $\forall s \in S$, $\forall k \in TP$. We denote the generic travel hop as an arc $e_i$ represented by $[(s, k), (s', k')]$, where $(s, k)$ and $(s', k')$ are the tail and the head of the arc, respectively. Each arc also has an associated weight $W(e_i)$ and a subscript $l$ that denotes one of the four possibilities:

- **Case 1.** wait arc $[(s, k), (s, k+1)]_{\text{WAIT}}$, where $s$ is the **Start stop** of a given package. This kind of arcs refers to the behavior of a package, which waits at its **Start stop** when the $k$th trip of the selected route comes. Here, the weight is given by the utilized time caused by waiting for the next trip, i.e., $W(e_i) = t_s^{k+1} - t_s^k$;

- **Case 2.** ride arc $[(s, k), (s+1, k)]_{\text{RIDE}}$. This kind of arcs models the behavior of a package riding a bus trip to its next stop. Here, the weight is given by the consumed time by riding the selected trip between two adjacent bus stops, i.e., $W(e_i) = t_s^k - t_s^{k-1}$;

- **Case 3.** re-wait arc $[(s, k), (s, k+1)]_{\text{REWAIT}}$. This kind of arcs models the behavior of a package that is being unloaded at bus stop $s$ at busy time of trip $k$, and then re-waiting at that stop for the next upcoming trip $k+1$. Here, the weight is given by the utilized time caused by re-waiting for the next trip, i.e., $W(e_i) = t_s^{k+1} - t_s^k$;

- **Case 4.** unload arc $[(s, k), (s+1, k)]_{\text{UNLOAD}}$, where $s$ is the **Target stop** in the package delivering. This kind of
Arcs models the behavior of a package having been unloaded at its Target stop, and would be further transmitted by city freighters. Here, the package delivering process has been terminated, i.e., $W(e) = 0$.

Fig. 3 shows an example of a multi-period directed multi-graph built according to the rules above. Specifically, package $p_i$ aims to be delivered from its Start stop $o_i$ to its Target stop $d_i$. The package initially waits at bus stop $o_i$, when an under-utilized trip $k_1$ comes, it rides that trip to the following bus stops. If the trip suffers from capacity shortage, the package would be unloaded at intermediate bus stops and re-wait for the following trips, for fear of impacting passengers. Finally, the package is unloaded at bus stop $d_i$ from trip $k_2$, where $k_2 \geq k_1$. Note that, a delivering path of a package can be presented as a continuous fold line and each sub-fold line directs right or down, because of the time sequence of the bus trips and the direction of the selected routes.

We model the action selection process of each package $p_i$ through the following non-splittable flow variables:

$$
\Phi^p_{[i,k),(s'k')]_{i}} = \begin{cases} 
1 & \text{if } p_i \text{ uses arc } [(s,k),(s',k')]_{i} \\
0 & \text{otherwise}.
\end{cases}
$$

Therefore, each arc in the multi-graph corresponds with one of the four kinds of states. For example, for ride arcs, we have

$$
\sum_{\{s,k\}\in S} \sum_{\{s',k\}\in S} \Phi^p_{[i,k),(s'k')]_{i}} \leq \sum_{s'k} v_{p_i,s,k}, \forall p_i \in P, s'k \in S \\
\sum_{\{s,k\}\in S} \sum_{\{s',k\}\in S} \Phi^p_{[i,k),(s'k')]_{i}} \leq \sum_{s'k} z_{p_i,s,k}, \forall p_i \in P, s'k \in S \\
\sum_{\{s,k\}\in S} \sum_{\{s',k\}\in S} \Phi^p_{[i,k),(s'k')]_{i}} \leq \sum_{s'k} x_{p_i,s,k}, \forall p_i \in P, s'k \in S \\
\sum_{\{s,k\}\in S} \sum_{\{s',k\}\in S} \Phi^p_{[i,k),(s'k')]_{i}} \leq \sum_{s'k} w_{p_i,s,k}, \forall p_i \in P, s'k \in S
$$

The objective function (8) aims to minimize the consumed time for delivering the packages, i.e., starting from the first package is loaded on a trip and ending at all packages are unloaded at their Target stops. The constraints could be explained as follows: Constraint (9) ensures that each package $p_i$ is delivered between any two adjacent stops by exactly one bus trip. Constraint (10) says that at any time slot $T_k^{i} \in \tilde{T}$, package $p_i$ only exhibits at most one state. Constraints (11)-(14) declare the conversion relationships between the four package states: after waiting at the Start stop $o_i$, package $p_i$ would ride an under-utilized trip $k'$ to the following bus stops; arriving stop $s+1$ by riding trip $k$, packages would still ride that trip or be unloaded; re-waiting at stop $s$ when trip $k+1$ arrives, packages would still re-wait, or ride that trip; after riding a trip to its Target stop, the state of the package would be fixed as being unloaded; Constraint (15) confines the volume of loaded packages when any trip $k$ travels from stop $s$ to stop $s+1$, considering the limited idle capacity. Constraint (16) states that after being unloaded at time $T_k^{i}$, the state of that package would not change at its next time slot, i.e., $T_{k+1}^{i}$ or $T_k^{i+1}$. Constraint (17) states that variable $w_{s,k}$ is only active when all packages are unloaded; (18) ensures that when the delivery task has been accomplished, the states of any packages would remain unchanged. Finally, constraint (19) defines the domains of the variables.

C. An efficient heuristic solution of the MILP-based model

The package delivering problem represents a complicated variant of an unsplitable multi-period multimmodity network design problem with side constraints, which is known to be NP-hard [22]. In order to further reduce the computational complexity, we propose an efficient heuristic approach that determines a suboptimal solution in polynomial time. Specifically, the heuristic solution can be addressed as a path planning problem using our constructed multi-period multimmodity network. Each package should pick a continuous fold line which starts from its Start stop and ends at its Target stop. The following designed rationales would facilitate our understanding of the process in determining the fold lines for all packages.

Rationale 1: When an under-utilized trip arrives, packages would be loaded in a determined sequence until the loaded amount reaches the upper bound. In this situation, the idle capacity of any trip could be fully exploited and the package...
Algorithm 1 Our efficient heuristic solution of the MILP-based model

1: for each arc \([s, k], (s', k')\) in \(G(V, A)\) do
2: initialize the arc weight
3: initialize the idle capacity \(Q^s_k\)
4: end for
5: for each package \(p_i\) in \(P\) do
6: initialize the package volume
7: determine the corresponding selected route
8: end for
9: while \(\Delta results > \sigma\) do
10: determine package priorities by the Genetic algorithm
11: for \(i = 1\) to \(|P|\) do
12: \(selected\text{trip} = 1\)
13: while any \(j\) in \((o_i\) to \(d_i)\) so that \(V(j, selected\text{trip}) + v_{p_i} \geq Q^s_{k_{selected\text{trip}}}\) do
14: \(selected\text{trip} = selected\text{trip} + 1\)
15: end while
16: update the loaded package volume \(V(s, k)\)
17: end for
18: result = \(\max\{selected\text{trip}\}\) of the \(|P|\) packages
19: end while
20: return solution

The solution would have a great scalability in the number of packages.

Rationale 2: Re-waiting stage should be avoided to the greatest extent. The re-waiting stage incurs more load and unload, which increases the input of labor resources and the risk of package loss. Therefore, we propose a rational that re-waiting stage should be avoided for work convenience and package security.

With such rationales, we design our solution as Algorithm 1 shows. Firstly, the traffic condition should be collected and evaluated (line 1 - 4). Then the packages would be generated with their volumes and locations. Then the selected bus route would be determined (line 5 - 8). The package loading priority, which incurs the most efficient package distribution, is determined through the Genetic algorithm [23] (line 10). In the package delivering process, all ranked packages wait at the Start stops until their selected trip arrives. The selected trip of package \(p_i\) is determined as the one that still has idle capacity from its Start stop \(o_i\) to its Target stop \(d_i\). Thereafter, the loaded volume of packages and idle capacity would be updated for each stop and each trip. In this way, the suboptimal solution of our CPDCP problem is found when \(\Delta results\) is below the given threshold. (line 9 - 20).

D. Model expansion to the whole traffic network

The solution can also be extended into the whole traffic network with a great number of bus routes. The first kind of extension is involving more bus routes into the CPTS. The second kind of extension is considering transfers between any two routes. Transfers are necessary when the first/last mile is relatively long utilizing only one route. The transfer stop of any two routes is fixed as the one that has the shortest distance to the other route. As Figure 5 shows, the packages would be unloaded at the Transfer stop#1, then be transmitted to the Transfer stop#2, finally wait there for upcoming trips of ROUTE#2. The time consumption of the transfer caused by 1) transferring; 2) re-waiting for the upcoming trip of ROUTE#2. In this way, any connected routes with transfers can be thought as one route. Thus, the package distribution in the whole traffic network can also be realized by our proposed efficient heuristic solution.

V. PERFORMANCE EMULATION

In this section, we empirically evaluate the performance of CPDCP. At first, we introduce the experimental settings used for demonstration in Section. V-A, including the experimental data, comparison algorithms, and evaluation metrics. Then we conduct performance emulations in Section V-B.

A. Experimental settings

1) Experimental Data: We use the real-world datasets for the evaluation, i.e., the road network data and the traffic data in the city of Changsha, China. This data is readily accessible using the published API from most city-run transportation systems. Since the datasets do not provide any information about package deliveries and the real-time passenger flow, we apply different mechanisms to emulate these two kinds of information. To emulate a package delivery task, we randomly generate a set of packages \((K)\), with their Original locations and Final destinations being determined based on the population heat map (see Fig. 4(a)). To emulate the passenger flow, we randomly generate the number of passengers waiting at each evolved bus stop using the Poisson distribution. In a CPTS vehicle, we assume the number of seats is 25. Each passenger occupies 1 seat. The volume of packages \((v_k)\) is generated with a uniform distribution, whose average value is 1/4 seat. \(\Delta\), the system parameters related to the Genetic algorithm, is set as \(\Delta = 1\). When the \(\Delta results\) < 1, the loop iteration terminates.

2) Comparison Algorithms and Evaluation Metrics: We complemented and compared the following algorithms:

No-rewaiting method (NRW). In this algorithm, no re-waiting stage is permitted. The priorities in loading packages are determined by the Genetic Algorithm.

Rewait-permitted methods with several priority policies. In this algorithm, the packages could be unloaded at the intermediate bus stops along the trip. The priority policies are:
- Start-from-departure-stop policy (R-SD): packages are firstly delivered to the departure stops of the selected bus routes before being distributed.
• Genetic algorithm policy (R-GA): packages are loaded in a priority determined by the Genetic Algorithm.
• Pri-distance policy (R-PD): packages with short delivering distance have higher loading priority.
• Pri-start-stop policy (R-SS): packages with a closer Start stop to the departure stop have higher loading priority.

We rigorously analyze the above methods using the following performance metrics.

**Delivering Time (DT)** Given a package delivering task, the delivering time starting from the first package being loaded from its Start stop and ending at the last package being unloaded to its Target stop. In this paper, we convert DT into the number of utilized trips for a package delivery task.

**Package State Distribution (PSD)** The PSD refers to the average state distribution of the packages in a package delivering task. The states of the delivering process include three parts: waiting at the original bus stop, re-waiting for upcoming trips, and riding the under-utilized trip.

The evaluation methodology is represented as follows. First, we generate parameters as described in Section V-A1 and then build the traffic (ξ) and passenger (Q) matrices. Given a set of packages, we plan package loading schemes based on our comparison algorithms, separately. Finally, we emulate the performance from two aspects, i.e., DT and PSD.

### B. Performance analysis

In this subsection, we conduct large-scale experiments to evaluate our CPDCP methods. We only extract traffic data at 12:00-12:30 am, since the bus trips at off-peak hours offer more idle capacity for package delivering. What’s more, all of the experimental values are the average of 1000 repeated experiments.

1) Preliminary experimental studies: Firstly, we do some preliminary experimental studies in terms of package coverage ratio and travel distance distribution.

Fig. 4(a) plots the population heat map at 10:00 pm in Changsha, China. Generally, customers send packages from their home. Therefore, the population distribution at 10:00 pm, when most people are at home, could represent the package distribution. Therefore, we generated the packages with their Original locations and Final destinations using the population heat map. The blocks/streets with more population would trigger more packages.

Fig. 4(b) shows the package coverage ratio under variable visible distance. In our tests, the packages are collected from the nearby streets and blocks within a visible distance. The visible distance is defined as the covering range of any bus stop. The packages can be delivered in our CPDCP mode when its Original location is within the coverage range from a bus stop. Fig. 4(b) shows that the package coverage ratio grows up with the increase of visible distance. Moreover, when the number of utilized routes increases, more packages could be delivered by the Crowdsourced Public Transportation Systems.

Fig. 4(c) shows the travel distance distributions with different permitted times of transfer. The travel distance of any package consists of three parts: 1) ride, the distance when the package on any bus; 2) transfer, the distance between the transfer stop of ROUTE#1 to the transfer stop of ROUTE#2; 3) First and Last mile (F&L mile), the distance of a package being transmitted from its Original location/Final destination to Start stop/Target stop. When the transfer is not permitted, the F&L mile is the longest. It is because that one package could only utilize one specific route in this mechanism, which limits the coverage scale. Additionally, transfer-permitted mechanism consumes more resource in the riding process. It is because that the transfer only appears between the dedicated transfer stops, which requires the buses to ride more distance for arriving the transfer stops.

2) DT: the number of utilized trips: Then, we emulate the performance in terms of the number of utilized trips.

Fig. 6(a) plots the number of utilized trips, with respect to the number of involved routes. Note that, there are hundreds of bus routes in Changsha, but only part of them could be involved in our CPDCP model, considering the feasibility and effectiveness. This figure plots the routes with the highest coverage rate. Given the package delivering task with 1000 randomly generated packages, fewer trips would be utilized for each route on average, as the number of involved routes increases. It is because that the packages would be allocated to various routes. This decreases the resource requirement and workload for each evolved route.

Fig. 6(b) plots the number of utilized trips for each route, when there are 10 selected Crowdsourced routes in total. The 10 Crowdsourced routes are selected with the highest coverage ratio. In these 10 selected bus routes, the first three ranked routes undertake about 88% packages, due to their
predominant locations and reactions. Other routes are only responsible for delivering about 12% packages.

Fig. 6(c) plots the number of utilized trips (average, max, and min) of the 10 selected routes, with respect to the number of packages in the delivering task. When the delivering task contains more packages, more capacity resource in each utilized route is demanded, leading to a larger number of utilized trips.

3) PSD: package state distribution: Finally, we compare the performance in terms of state distribution.

Fig. 7(a) plots the PSD of several proposed methods, with respect to the number of involved routes ranging from 2 to 10. With the number of selected routes increases, the average utilized time for delivering a package decreases gradually. It is because the package delivering task is allocated to more routes, thus the workload of each route is relieved. In the R-SD method, because all packages are delivered to the departure stop of the corresponding route firstly, 1) the time at wait stage is extremely short, because the packages can be loaded when a trip starts to operate; 2) the total utilized time increases a lot, due to the longer delivering distance. For the re-wait-permitted methods with different priority schemes, the total utilized time is longer than that of NRW. It is because that in re-wait-permitted methods, more packages would be loaded for each trip because that the upper bound of the loaded amount is only confined by the idle capacity of the Start stop. However, these packages have a high probability to be unloaded at the following stops for ensuring passenger experience. Therefore, the re-waiting stage would be extremely long waiting for the next trips.

Fig. 7(b) plots PSD of several proposed methods with respect to the number of packages ranging from 200 to 1000. With the number of packages increases, the average utilized time for delivering a package increases, due to more demand for capacity resources. Moreover, for re-wait-permitted policies, the re-wait stage occupies more time when the delivering task contains more packages, because of the capacity resource contention.

In summary, the package delivering task can be accomplished satisfactorily by our proposed CPDCP model. More than 60% packages could be delivered to their Final destinations within 500m around of their dedicated routes, when the CPTS involves 10 routes. Given 1000 packages to be delivered in off-peak working hours, our heuristic method only occupies 7.5 trips on average and utilizes less than 4 hours to accomplish the delivering task, while other methods consume more resources and time consumption.

VI. CONCLUSION AND FUTURE WORK

In this paper, we present a novel framework called CPDCP. The framework exploits the under-utilized capacity of CPTS vehicles to deliver packages. We first identify the four states in package delivering, i.e., waiting, riding, re-waiting, and being unloaded. Thereafter, we carefully plan the delivering scheme by determining the states of any package at any time slot, to minimize the utilized time for delivering all packages. We mathematically formulate this problem using Mixed Integer Linear Programming techniques and propose an efficient heuristic solution for this NP-hard problem. In the evaluation, we compare our proposed methods and find that our method accomplishes the package delivering task in a more effective and efficient way.
This paper is the first attempt to investigate the package loading problem under the CPDCP framework. The research topic can be further extended in several directions. First, some design rationales for package delivering would be revisited. For instance, we assume that package should avoid re-waiting for less labor resource input and package loss in the heuristic solution, but this assumption may not gain the highest efficiency in accomplishing the package delivery tasks. Second, our emulation is based on the traffic data in off-peak hours. It is meaningful to update the live traffic in further studies, which is much closer to the practical application. Finally, it is an important extension for the real-world case study and benchmark instance design, which can be used to test different models and methods in this field.

ACKNOWLEDGMENT

This work is partially supported by National Natural Science Foundation of China under Grant No.61772544, National Basic Research Program (973 program) under Grant No.2014CB347800, the Hunan Provincial Natural Science Fund for Distinguished Young Scholars under Grant No.2016JJ1002, and the Guangxi Cooperative Innovation Center of cloud computing and Big Data under Grant Nos.YD16507 and YD17X11.

REFERENCES


[8] Q. M. Ha, Y. Deville, Q. D. Pham, and M. H. H., "On the min-cost traveling salesman problem with drone."


Geyao Cheng
Geyao Cheng received the B.S. degree in management science and engineering from National University of Defense Technology, Changsha, China, in 2016. She is currently working towards the M.S. degree in College of Systems Engineering, National University of Defense Technology, Changsha, China. Her research interests include mobile computing and crowdsensing.

Deke Guo
Deke Guo (S06CM08) received the B.S. degree in industry engineering from the Beijing University of Aeronautics and Astronautics, Beijing, China, in 2001, and the Ph.D. degree in management science and engineering from the National University of Defense Technology, Changsha, China, in 2008. He is currently a Professor with the College of Systems Engineering, National University of Defense Technology. His research interests include distributed systems, software-defined networking, data center networking, wireless and mobile systems, and interconnection networks. He is a member of the ACM.

Jianmai Shi
Jianmai Shi received the Ph.D. degree in management science from National University of Defense Technology (NUDT), Changsha, China, in 2011. He is currently an associate professor in college of system engineering, NUDT. His main research interests include heuristic algorithm design, reverse logistic, vehicle routing problems and supply chain management.

Yudong Qin
Yudong Qin received the B.S. degree in management science and engineering from National University of Defense Technology, Changsha, China, in 2016. He is currently working towards the M.S. degree in College of Systems Engineering, National University of Defense Technology, Changsha, China. His research interests include data centers and software-defined networks.