

An Augmented Estimation of Distribution Algorithm For Multi-Carpooling Problem With Time Window

Fang Zhang

School of Computer and
Communication Engineering, Liaoning
Shihua University
Fushun 113001, Liaoning, China
zhangfang@lnpu.edu.cn

Z.J. Yang, Y. Wang[†], Member, IEEE

Lab. of Networked Control Systems,
Shenyang Institute of Automation,
Chinese Academy of Sciences
Shenyang 110016, Liaoning, China
{yang, wangyang}@sia.ac.cn

Fangjun Kuang

University of Stuttgart
Pfaffenwaldring 47, 70569 Stuttgart,
Germany
kuangfn@studi.informatik.uni-
stuttgart.de

Abstract—A multi-carpooling model is proposed for the multi-vehicle carpooling problem in distributed parallel computing environment. A two-stage stochastic optimization of the estimation of distribution algorithm solves the optimum of the multi-carpooling problem with a carpooling probabilistic matrix. A rideable matrix initiates the carpooling probabilistic matrix, and the carpooling probabilistic matrix continues updating during the optimization. The carpooling model mines efficient and compromised ridesharing routes for shared riders by the optimization iterations. Experimental results indicate that the carpooling model has the characteristics of effective and efficient traffic including shorter waiting time, more passenger load, and less average riding distance.

Keywords—carpooling problem; pickup and delivery problem; estimation of distribution algorithm; stochastic optimization.

I. INTRODUCTION

The multi-carpooling problem is a NP sub-problem of Pickup and Delivery Problem which goal is to minimize the riding distances and thus alleviate urban traffic jams and the transportation pressure^[1,2]. Meanwhile it subjects to the constraints of multi-variables including: location points of dynamic ridesharing, time windows for pickup and dropping, riding routes, passenger capacities, travel speed and etc. The implementation of the carpooling problem is to seek compromised multi-routes that are preferred by all qualified riders; however, stochastic disturbances of travel time make it hard to actionable merging the routes for all riders. Yan proposed a stochastic carpooling model formulated as an integer multiple commodity network flow problem; however, the model could be difficult to find optimal solutions especially with large problem sizes^[3]. A more conventional approach is to use vehicle routing problem with hard time window to eliminate the uncertainty^[4,5]. In addition, the multi-carpooling model is implemented in the multi-vehicle environment with distributed parallel computing^[6-9]. The distributed computing for multi-carpooling is of high time and space complexity, hence, the temporal-spatial complexity results in the solving difficulties within a reasonable period of time. In order to reduce the complexity, Pan^[4] proposed a time difference insertion heuristics algorithm, Shao^[9] proposed a clustering heuristic algorithm based on matching degree, and Xiao^[10]

used the fuzzy clustering and fuzzy recognition theory to study the carpooling problem. Though their heuristic algorithms are efficient, yet these models were simplified by a certain level of abstraction, the risk remains that the model would not represent reality sufficiently. As for the model-free method, service models^[6-8] are proposed as a data-driven approach considering GPS data sets, context information, and measurement based on the MapReduce model. Besides, genetic algorithm^[5,11] and ant colony algorithm^[12] are also be used to solve the carpooling problems in cloud computing.

For more efficient computing and effective solutions, an estimation of distribution algorithm (sometimes called probabilistic model-building genetic algorithm) is proposed as a stochastic optimization method to solve the optimum with a carpooling probabilistic matrix of promising carpooling solutions. The optimization consists of a series of incremental updates of the carpooling probabilistic matrix. If the carpooling probabilistic matrix is not the identity matrix \mathbf{I} , the multi-carpooling problem is augmented with an intermediate stage for ridesharing among drivers. In addition, a notion of the virtual vehicle is introduced as a generalized profile of vehicles permitting arbitrary starting and ending points for drivers.

II. PROBLEM FORMULIZATION

A. Definition of carpooling model

Definition 1 The topology of urban transportation networks is denoted by the graph $g(\mathcal{V}, \mathcal{E}, \mathcal{W})$. Assuming that there are m drivers and n passengers, m drivers only accept self-driving travel and ridesharing travel, and n passengers only accept ridesharing travel and bus travel. Each vehicle keeps moving at a constant speed. The service-based multi-carpooling model automatically identifies the m vehicles which are recommended for carpooling services, which provide ridesharing for qualified riders ($\leq m+n$) in multiple pickup and dropping points.

Definition 2 The service time window specifies the waiting time interval $[e_x, l_x]$ for the pickup and dropping point x . All vehicles have to pick up and drop down riders in the specified time window. When a vehicle reaches a certain point x at the time $t_{A(x)}^s$, the waiting time is denoted by $t_{W(x)}^s = \max\{e_x - t_{A(x)}^s, 0\}$.

This work was supported by the Natural Science Foundation of China under contract (61233007), the National High Technology Research and Development Program of China (863 Program: 2011AA040101, 2012AA041102-03, 2011AA040103), Foundation of Chinese Academy of Sciences under contract (XDA06020500, XDA06021100, XDA06020602).

Corresponding author: yang.wang.cn@ieee.org

Definition 3 The virtual vehicle $\mathcal{S} = \{s_j \mid j \in [1, m]\}$ gives a generalized profile for the distributed vehicles which could be not recommended for carpooling services: If a virtual vehicle s is recommended, then $\mathcal{I}_{sd^+}^s = \mathcal{I}_{sd^-}^s = 1$, If a virtual vehicle s is not recommended, then $\mathcal{I}_{sd^+}^s = \mathcal{I}_{sd^-}^s = 0$; , where s_j denotes the index of vehicles, and the subscript j denotes the dynamic driver who could be not the car owner. The initial value and the final value of \mathcal{S} are both $\{s_j \mid j=1, 2, \dots, m\}$.

TABLE I. NOTATION NOMENCLATURE

g	Graph of urban transportation network $g(\mathcal{V}, \mathcal{E}, \mathcal{W})$
\mathcal{V}	Set of urban crossing points
\mathcal{E}	Set of urban transportation lines
\mathcal{W}	Adjacency matrix $\mathcal{W} = [w_{xy} \mid x, y \in \mathcal{V}]$, w_{xy} denotes the distance
m	Index of drivers $i \in [1, m]$
n	Index of passengers $i \in [1, n]$
\mathcal{D}^+	Set of drivers' starting points d_i^+ , $\mathcal{D}^+ = \{d_i^+ \mid i \in [1, m]\}$
\mathcal{D}^-	Set of drivers' ending points d_i^- , $\mathcal{D}^- = \{d_i^- \mid i \in [1, m]\}$
\mathcal{D}	Set of drivers' travel routes d_i , $\mathcal{D} = \{d_i^+ d_i^- \mid i \in [1, m]\}$
\mathcal{P}^+	Set of passengers' starting points p_i^+ , $\mathcal{P}^+ = \{p_i^+ \mid i \in [1, n]\}$
\mathcal{P}^-	Set of passengers' ending points p_i^- , $\mathcal{P}^- = \{p_i^- \mid i \in [1, n]\}$
\mathcal{P}	Set of passengers' travel routes p_i , $\mathcal{P} = \{p_i^+ p_i^- \mid i \in [1, n]\}$
\mathcal{A}^+	Set of all starting points a_i^+ , $i \in [1, m+n]$, $\mathcal{A}^+ = \mathcal{D}^+ \cup \mathcal{P}^+$
\mathcal{A}^-	Set of all ending points a_i^- , $i \in [1, m+n]$, $\mathcal{A}^- = \mathcal{D}^- \cup \mathcal{P}^-$
\mathcal{A}	Set of travel routes a_i , $\mathcal{A} = \{a_i^+ a_i^- \mid i \in [1, m+n]\}$
\mathcal{S}	Set of drivers' virtual vehicle s_j , $\mathcal{S} = \{s_j \mid j \in [1, m]\}$
\mathcal{I}_{xy}^s	Binary value is 1 if vehicle s traveling from point x to y , else 0
Q^s	Passenger capacity of vehicle s
q_x^s	Passenger load of vehicle s arriving at point x
$q_{x^+}^s$	Number of passengers getting on vehicle s arriving at point x
$q_{x^-}^s$	Number of passengers getting off vehicle s arriving at point x
v_s	Speed of vehicle s
t_{xy}^s	Travel time $t_{xy}^s = w_{xy} / v_s$ of vehicle s traveling from point x to y
$t_{A(x)}^s$	Arriving time of vehicle s arriving at point x
$t_{L(x)}^s$	Leaving time of vehicle s leaving at point x
e_x	Lower bound of the time window at point x
l_x	Upper bound of the time window at point x
$t_{W(x)}^s$	Waiting time of vehicle s at point x
$t_{R(a)}^s$	Riding time for a rider in vehicle s
R	Ridable matrix with $(m+n) \times (m+1)$ dimensions
P	Carpooling probabilistic matrix with $(m+n) \times (m+1)$ dimensions
\mathcal{F}	Fitness function $\mathcal{F}(g, \mathcal{S}, \mathcal{A})$
ω	Inertia coefficient
κ	Number of candidate solutions in each generation

B. Objectives of carpooling model

Objective 1 Minimum the number of vehicles for all the drivers' travel:
$$\min \sum_{s \in \mathcal{S}} \bigcup_{d \in \mathcal{D}} \mathcal{I}_d^s, d \in \mathcal{D}, s \in \mathcal{S} \quad (1)$$

, where \sum denotes the cumulative operator for all recommended vehicles, and \cup denotes the union operator for the carpooling of the drivers' ridesharing routes.

Objective 2 Maximum the number of riders for ridesharing travel:
$$\max \sum_{a \in \mathcal{A}} \bigcup_{s \in \mathcal{S}} \mathcal{I}_a^s, a \in \mathcal{A}, s \in \mathcal{S} \quad (2)$$

, where \sum denotes the cumulative operator for all qualified riders, and \cup denotes the union operator for the ridesharing routes of all recommended vehicles.

C. Constraints of carpooling model

Constraint 1 for m drivers-All of the m drivers must be qualified as riders:
$$\sum_{s \in \mathcal{S}} \mathcal{I}_d^s = 1, \forall d \in \mathcal{D} \quad (3)$$

Constraint 2 for virtual vehicles-All of recommended vehicles must be driven at the starting and ending points by the correlated car owners:
$$\mathcal{I}_{sd^+}^s \cdot \mathcal{I}_{sd^-}^s = 1, \exists s \in \mathcal{S} \quad (4)$$

Constraint 3 for the time window-All of vehicles must arrive at the pickup and dropping point x before upper bound of the time window:
$$t_{A(x)}^s \leq l_x, \forall x \in \mathcal{V} \quad (5)$$

Constraint 4 for the passenger capacity-The passenger load of all vehicles cannot be more than the passenger capacity at all points:
$$q_x^s \leq Q^s, \forall s \in \mathcal{S}, \forall x \in \mathcal{V} \quad (6)$$

Constraint 5 for pickup and dropping points-Every riders can get on and off at the pickup and dropping point x only once:
$$\sum_{s \in \mathcal{S}} \mathcal{I}_{a^+x}^s \cdot \sum_{s \in \mathcal{S}} \mathcal{I}_{xa^-}^s = 1, \forall x \in \mathcal{V} \quad (7)$$

Constraint 6 for the ending point-The time for starting plus riding cannot be more than upper bound of the time window at the ending point:
$$t_{A(a^+)}^s + t_{R(a)}^s \leq l_{a^-}, \forall a^- \in \mathcal{A}^- \quad (8)$$

Constraint 7 for the waiting time-When a vehicle reaches a certain point x at the time $t_{A(x)}^s$, if the service time window is still close (namely $t_{A(x)}^s < e_x$), then the vehicle has to wait for the lower bound of the time window until the leaving time:
$$t_{L(x)}^s = \max\{t_{L(x)}^s, e_x\}, \exists t_{A(x)}^s < e_x \quad (9)$$

Constraint 8 for the riding time-The riding time cannot be more than the time from starting point to ending point, because of waiting time:
$$t_{R(a)}^s \leq t_{A(a^-)}^s - t_{L(a^+)}^s, \forall a \in \mathcal{A} \quad (10)$$

III. STOCHASTIC OPTIMIZATION

In order to solve the objectives (1) and (2) subjected to the constraints (3)-(10), a stochastic optimization method based

on a two-stage estimation of distribution algorithm (EDA for short) is proposed.

A. Ridable matrix

EDAs always use an explicit probability to reach a faster convergence than the implicit probability of most conventional evolutionary algorithms; however, identity matrix \mathbf{I} usually used for initializing probabilistic matrix is not suitable for the multi-carpooling optimization. Since the identity matrix \mathbf{I} will results in a large number of infeasible solutions especially in the initial stage of evolutionary computing, a ridable matrix \mathbf{R} is proposed for improving the quality of initial probabilistic matrix. The ridable matrix \mathbf{R} makes full use of the information including objective (1), objective (2), and constraints (3)-(10).

Definition 4 The ridable matrix $\mathbf{R}_{(m+n) \times (m+1)}$ denotes the relationship whether the $m+n$ riders are able to ride the m vehicles or the 1 bus, where i, j entry of $\mathbf{R}_{(m+n) \times (m+1)}$ is setup as:

$$\mathbf{R}_{ij} = \begin{cases} 1, & \text{for } j \in \{1, \dots, m\}, \text{ if constraints (3)-(10) can hold for rider } i \\ 0, & \text{for } j \in \{1, \dots, m\}, \text{ If constraints (3)-(10) cannot hold for rider } i \\ 1, & \text{for } j = m+1, \text{ If the rider is a passenger } i \in \{m+1, \dots, m+n\} \\ 0, & \text{for } j = m+1, \text{ If the rider is a driver } i \in \{1, 2, \dots, m\} \end{cases}$$

B. Carpooling probabilistic matrix

Definition 5 The carpooling probabilistic matrix $\mathbf{P}_{(m+n) \times (m+1)}$ denotes the probability that the $m+n$ riders ride the m vehicles and the bus, where the (i, j) entry of $\mathbf{P}_{(m+n) \times (m+1)}$ denotes the probability of the rider i picked up by the vehicle j .

The initial carpooling probabilistic matrix $\mathbf{P}^{(0)}$ is defined as:

$$\mathbf{P}_{ij}^{(0)} = \begin{cases} \mathbf{R}_{ij} / \sum_{j=1}^{m+1} \mathbf{R}_{ij}, & \text{if } \sum_{j=1}^{m+1} \mathbf{R}_{ij} \neq 0 \\ 0, & \text{if } \sum_{j=1}^{m+1} \mathbf{R}_{ij} = 0 \end{cases}$$

The update of carpooling probabilistic matrix $\mathbf{P}^{(k)}$ is defined as:

$$\mathbf{P}_{ij}^{(k)} = \omega \mathbf{P}_{ij}^{(k-1)} + (1 - \omega) \sum_{\kappa=1}^{\kappa} \mathbf{R}_{ij}^{(\kappa)} / \kappa \quad (11)$$

, where $\mathbf{P}_{ij}^{(k-1)}$ denotes the carpooling probabilistic of admissible solutions in the $k-1$ th generation, ω denotes the inertia coefficient, κ denotes the number of candidate solutions in the k th generation, $\mathbf{R}_{ij}^{(\kappa)}$ denotes whether exists one satisfying the vehicle j ridable by rider i of the κ candidate solutions in the k th generation.

C. Fitness Function

According to the objective (1) minimizing recommended vehicles satisfying all drivers' travel, and the objective (2) maximizing qualified riders satisfying most passengers' ridesharing travel, the fitness function also takes the minimum riding distance of all recommended vehicles into account:

$$\mathcal{F}(g, \mathcal{S}, \mathcal{A}) = \left(\alpha \sum_{s \in \mathcal{S}} \sum_{d \in \mathcal{D}} \mathcal{I}_d^s + \beta \left(m + n - \sum_{a \in \mathcal{A}} \sum_{s \in \mathcal{S}} \mathcal{I}_a^s \right) + \gamma \sum_{s \in \mathcal{S}} \sum_{x, y \in \mathcal{V}} w_{xy} \mathcal{I}_{xy}^s \right)^{-1} \quad (12)$$

D. Augmented EDA

Augmented EDA consists of first stage and second stage.

Stage 1 is to implement ridesharing among drivers, namely, objective (1) minimizing recommended vehicles satisfying all drivers' travel.

TABLE II. FIRST STAGE OF AUGMENTED EDA

1	Initialize the ridable matrix \mathbf{R} in Definition 4 and the carpooling probabilistic matrix $\mathbf{P}^{(0)}$ in Definition 5 . Set $k=1, \kappa=0$.
2	Clear all the ridesharing routes, and set all the routes $\mathcal{I}_{xy}^s = 0$; do not recommend any vehicle $\mathcal{I}_{sd^+}^s = 0, \mathcal{I}_{sd^-}^s = 0$.
3	Randomly select one driver $i \in [1, m]$ who has not a qualified ridesharing route. Allocate a feasible virtual car j for driver i 's travel $d_i \in \mathcal{D}$ with the carpooling probability $\mathbf{P}_{ij}^{(k-1)}$.
4	If the driver i selects his own vehicle, then recommend this vehicle, and set $\mathcal{I}_{sd^+}^s = \mathcal{I}_{sd^-}^s = 1$; meanwhile, allocate a qualified ridesharing route for driver i 's travel $d_i \in \mathcal{D}$, and set $\mathcal{I}_d^s = 1$.
5	If the driver i has not a qualified ridesharing route, then allocate a qualified ridesharing route for driver i , and set $\mathcal{I}_d^s = 1$.
6	If any one driver has not a qualified ridesharing route, then goto 3.
7	If all the m drivers have qualified ridesharing routes, then generate a new candidate solution, set $\kappa=\kappa+1$. If $\kappa < N$, then goto 2.
8	Update the entries $\mathbf{P}_{ij}^{(k)}$ of carpooling probabilistic matrix \mathbf{P} using (11)
9	If termination criteria $\mathcal{F} \leq \varepsilon$ met, output; else set $k=k+1, \kappa=0$, goto 2.

Stage 2 is to utilize the recommended vehicles for qualifying most riders' ridesharing routes, namely, objective (2) maximizing qualified riders satisfying most riders' ridesharing travel.

TABLE III. SEACOND STAGE OF AUGMENTED EDA

1	Initialize the ridable matrix \mathbf{R} in Definition 4 and the carpooling probabilistic matrix $\mathbf{P}^{(0)}$ in Definition 5 . Set $k=1, \kappa=0$.
2	Initialize the output of first stage to the ridesharing routes, and make all riders not qualified $\mathcal{I}_a^s = 0$.
3	Randomly select one rider $i \in [1, m+n]$ who has not a qualified ridesharing route. Allocate a feasible virtual car j for rider i 's travel $a_i \in \mathcal{A}$ with the carpooling probability $\mathbf{P}_{ij}^{(k-1)}$.
4	If the rider i has not a qualified ridesharing route, then allocate a qualified ridesharing route for rider i , and set $\mathcal{I}_a^s = 1$.
5	If any one rider has not a qualified ridesharing route, then goto 3.
6	If all riders have qualified ridesharing routes, then generate a new candidate solution, set $\kappa=\kappa+1$. If $\kappa < N$, then goto 2.
7	Update the entries $\mathbf{P}_{ij}^{(k)}$ of carpooling probabilistic matrix \mathbf{P} using (11)
8	If termination criteria $\mathcal{F} \leq \varepsilon$ met, output; else set $k=k+1, \kappa=0$, goto 2.

E. Single-Stage EDA

Only making the carpooling probabilistic matrix \mathbf{P} identity matrix $\mathbf{I}_{(m+n) \times (m+1)}$, the augmented EDA with two stages will eliminate the first stage, and directly degrade to a single-stage EDA. In single-stage EDA, all vehicles are recommended.

TABLE IV. SINGLE STAGE EDA

- 1 Initialize the rideable matrix \mathbf{R} in **Definition 4** and the carpooling probabilistic matrix $\mathbf{P}^{(0)}$ in **Definition 5**. Set $k=1, \kappa=0$.
- 2 Clear all the ridesharing routes, do not recommend any vehicle, make all the riders not qualified, thus set all $\mathcal{I}_{xy}^s = \mathcal{I}_{sd}^s = \mathcal{I}_{sa}^s = \mathcal{I}_a^s = 0$.
- 3 Randomly select one rider $i \in [1, m+n]$ who has not a qualified ridesharing route. Allocate a feasible virtual car j for rider i 's travel $a_i \in \mathcal{A}$ with the carpooling probability $\mathbf{P}_{ij}^{(k-1)}$.
- 4 If the rider i selects his own vehicle, then recommend this vehicle, and set $\mathcal{I}_{sa}^s = \mathcal{I}_{sa}^s = 1$; meanwhile, allocate a qualified ridesharing route for driver i 's travel $a_i \in \mathcal{A}$, and set $\mathcal{I}_a^s = 1$.
- 5 If the rider i has not a qualified ridesharing route, then allocate a qualified ridesharing route for rider i , and set $\mathcal{I}_a^s = 1$.
- 6 If any one rider has not a qualified ridesharing route, then goto 3.
- 7 If all riders have qualified ridesharing routes, then generate a new candidate solution, set $\kappa=\kappa+1$.
If $\kappa < N$, then goto 2.
- 8 Update the entries $\mathbf{P}_{ij}^{(k)}$ of carpooling probabilistic matrix \mathbf{P} using (11)
- 9 If termination criteria $\mathcal{F} \leq \varepsilon$ met, output; else set $k=k+1, \kappa=0$, goto 2.

IV. NUMERIAL SIMULATION

We bought part travel data of Shenyang citizens from a tertiary commerce database. The data of 40 citizens' travel are randomly selected, including complete information from 20 drivers and 20 passengers. A visualization based on Java & Processing has been programmed^[8]. The comparison between non-carpooling (NC), first stage (FS), augmented EDA (AEDA), and single-stage EDA (SEDA) is incarnated according to the experimental results of the 40 travel data.

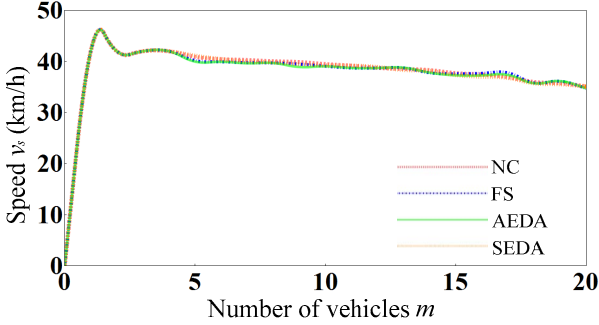


Fig. 1. Average riding speed

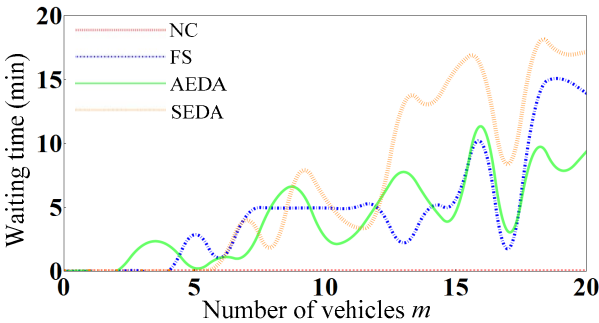


Fig. 2. Average waiting time

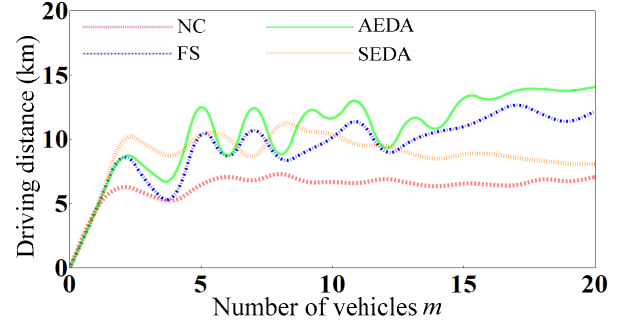


Fig. 3. Average driving distance per vehicle

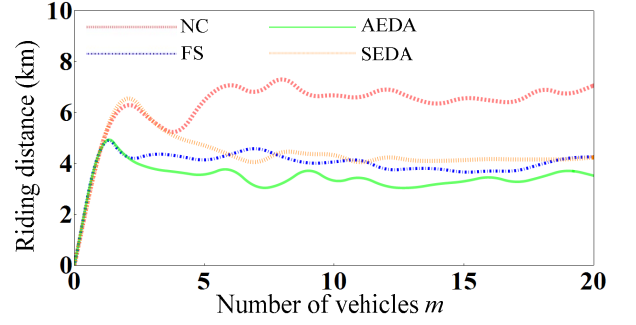


Fig. 4. Average riding distance per rider

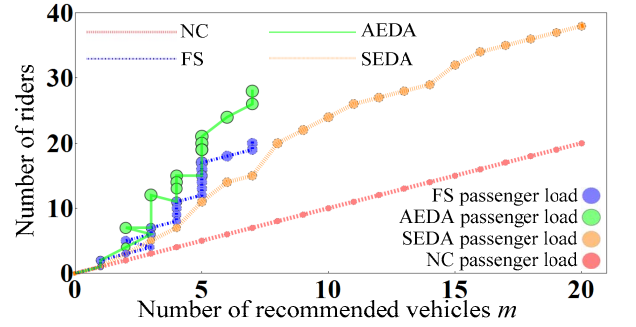


Fig. 5. Passenger load per vehicle and total number of rides

The difference of average riding speeds between NC, FS, AEDA, and SEDA is small (Fig.1), respectively 35.1204 km/h, 34.7317 km/h, 34.7233 km/h, and 35.0923 km/h at $m=20$. The average waiting time of NC is zero all the time, since NC need not wait anybody at any pickup and dropping point. The average waiting time of FS, AEDA, and SEDA is 13.914 min, 9.37765 min, and 17.1596 min at $m=20$. Namely, AEDA reduces 33% waiting time compared by FS, and 45% waiting time compared by SEDA (in Fig. 2).

Because of dynamic ridesharing, AEDA takes up more riders than FS, SEDA, and NC in descending order. The average driving distances per vehicle of NC, FS, AEDA, and SEDA are 7.0685 km, 12.1953 km, 14.0871 km, and 8.05993 km at $m=20$; however, the accumulative driving distances of NC, FS, AEDA, and SEDA are 141.37 km with 20 vehicles, 85.3671km with 7 vehicles, 98.6096 km with 7 vehicles, and 161.19855 km with 20 vehicles at $m=20$ (in Fig. 3). AEDA reduces 65% vehicles compared with NC and SEDA, while

AEDA picks up 40% more passengers with 16% more distance than FS.

Every rider's average riding distance represents the efficiency of the carpooling service. Obviously AEDA is the most efficient with 3.52177 km at $m=20$ (Fig. 4). The values of NC, FS, and SEDA are 7.0685 km, 4.26835 km, and 4.24207 km, which efficiency is 50%, 83%, and 83% by AEDA.

As shown in Fig. 5, AEDA minimizes the number of recommended vehicles, and maximizes the number of qualified rides. The accumulative passenger load of NC, FS, AEDA, and SEDA reaches 20 riders with 20 vehicles, 20 riders with 7 vehicles, 28 riders with 7 vehicles, and 38 riders with 20 vehicles. Average passenger load per vehicle is 1 rider for NC, 2 riders for SEDA, 3 riders for FS, and 4 riders for AEDA.

TABLE V. COMPARISON BETWEEN NC, FS, AEDA, AND SEDA

$m=20$	NC	FS	AEDA	SEDA
speed(km/h)	35.1204	34.7317	34.7233	35.0923
waiting time(min)	0.000	13.914	9.37765	17.1596
driving distances(km)	7.0685	12.1953	14.0871	8.0599
total distances(km)	141.37	85.3671	98.6096	161.1986
riding distance(km)	7.0685	4.26835	3.52177	4.2420
used vehicles	20	7	7	20
passenger load	1	3	4	2
total riders	20	20	28	38

Table V records comprehensive experimental results at $m=20$. The augmented estimation of distribution algorithm shows the most average driving distances per vehicle(Fig. 3), the least riding distance per rider(Fig. 4), the least recommended vehicles, and the most passenger load(Fig. 5).

Conclusion and Future Works

A multi-carpooling model based on the two-stage estimation of distributed estimation algorithm is proposed for smarter traffic in intelligent transportation. The experimental results prove that the model is efficient and effective, and the optimization method can greatly reduce the traffic jam and the pressure of public transportation. This model is of high flexibility and feasibility. By modifying the carpooling

probabilistic matrix of the augmented EDA, the model can be generalized into other carpooling models. The future work will focus on integrating the information of the urban transportation network and the vehicular dynamic routing to realize a carpooling recommendation system with the high performance of distributed optimization.

REFERENCES

- [1] Wen He, Kai Hwang, Deyi Li. "Intelligent Carpool Routing for Urban Ridesharing by Mining GPS Trajectories". IEEE Transactions on Intelligent Transportation Systems, 2014, 15(5):2286-2296.
- [2] Feng He, Sengbin Yu, Ruichun He, et al. "Research on optimization model of taxi-carpooling expenses based on the passengers' personalized demand". In Proceedings of International Conference on Transportation Information and Safety, 2015, pp:246-249.
- [3] Shangyao Yan, Chun-Ying Chen, Sheng-Chieh Chang. "A Car Pooling Model and Solution Method With Stochastic Vehicle Travel Times". IEEE Transactions on Intelligent Transportation Systems, 2014, 15(1):47-61.
- [4] Lijun Pan, Zhuo Fu, et al. "Time Difference Insertion Heuristics Algorithm for Vehicle Routing Problem With Hard Time Window". Journal of Computer Applications, 2012,32(11):3402-3403.
- [5] Jianqiang Zhang, Weiguo Fang. "Improved Genetic Algorithm for Vehicle Routing Problem With Time Window". Computer Engineering and Applications. 2010, 46(32):228-231.
- [6] Desheng Zhang, Tian He, Yunhuai Liu, et al. "A Carpooling Recommendation System for Taxicab Services". IEEE Transactions on Emerging Topics in Computing, 2014, 2(3):254-266.
- [7] Zhongmei Zhang, Guiling Wang, Bo Cao, et al. "Data Services for Carpooling Based on Large-Scale Traffic Data Analysis". In Proceedings of IEEE International Conference on Services Computing, 2015, pp: 672-679.
- [8] Zi Wang, Tong Zhou, Yang Wang, et al. "Design and Implementation of Intelligent Carpooling system based on GPS data visualization". Computer Engineering and Design, 2015, 36(2):529-533.
- [9] Zengzhen Shao, Hongguo Wang, et al. "Research on Service Requirement Distribution Algorithms in Carpooling Problems". Journal of Tsinghua University, 2013, 53(2):252-258.
- [10] Qiang Xiao, RuiChun He, Wei Zhang, et al. "Algorithm Research of Taxi Carpooling Based on Fuzzy Clustering and Fuzzy Recognition". Journal of Transportation Systems Engineering and Information Technology, 2014, 14(5):119-125.
- [11] Shih-Chia Huang, Ming-Kai Jiau, Chih-Hsiang Lin. "A Genetic-Algorithm-Based Approach to Solve Carpool Service Problems in Cloud Computing". IEEE Transactions on Intelligent Transportation Systems, 2015, 16(1):352-364.
- [12] Chunhua Meng, Hongguo Wang, et al. "Algorithm Research on Multi-Vehicle Ride Matching Problem Based on Passengers Classification and Transfers". Computer Science, 2013, 40(9):237-242.