This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

An Evolutionary Multiobjective Carpool Algorithm Using Set-Based Operator Based on Simulated Binary Crossover

Jing-Jie Lin, Shih-Chia Huang, *Senior Member, IEEE*, and Ming-Kai Jiau

***Abstract*—Sharing vehicle journeys with other passengers can provide many benefits, such as reducing traffic congestion and making urban transportation more environmentally friendly. For the procedure of sharing empty seats, we need to consider increased ridership and driving distances incurred by carpool detours resulting from matching passengers to drivers, as well as maximizing the number of simultaneous matches. In accordance with these goals, this paper proposes and defines the multiobjective optimization carpool service problem (MOCSP). Previous studies have used evolutionary algorithms by combining multiple objectives into a single objective through a weighted linear or/and nonlinear combination of different objectives, thus turning to a single-objective optimization problem. These single-objective problems are optimized, but there is no guarantee of the performance of the respective objectives. By improving the individual representation and genetic operation, we developed a set-based simulated binary and multiobjective carpool matching algorithm that can more effectively solve MOCSP. Furthermore, the proposed algorithm can provide better driver-passenger matching results than can the binary-coded and set-based nondominated sorting genetic algorithms.**

***Index Terms*—Evolutionary algorithm (EA), multiobjective problem, set-based coding.**

I. Introduction

CARPOOLING as a ride-sharing service not only cuts transportation costs and energy consumption but also reduces the number of vehicles on roadways, thereby easing traffic congestion in the transportation system. Effective ride-sharing must address the carpool service problem (CSP), in which available vehicle seats are matched with people who have similar itineraries. Specifically, the primary CSPs are concerned with matching two types of people who participate in carpooling activities: 1) "drivers" who have vacant seats in the vehicles and 2) "passengers" who are looking for rides. Numerous research works have studied the CSPs and provided various effective solutions for them. Lin *et al.* [1] proposed an intelligent carpool system and basic method that provides matching results, the processing times for which can be accelerated by increasing the proportion of randomly generated matching results to all possible matching results and thus dynamically adjusting the sample size.

The dial-a-ride problem (DARP) is generalizations of pickup and delivery problem with time windows aim to manage a fleet of vehicles to handle a set of users specifying conditions in term of pickup, delivery, and time windows. The door-to-door service for people who are older or disabled is an application of DARP. A genetic algorithm (GA) is proposed [2] in the view of service level constraints to optimize the total transportation cost. Also, models and algorithms are proposed [3] for the DARP under a heterogeneous fleet of vehicles and driver-relevant constraints. The CSP with additional restrictions that all drivers are limited to start their trip from the single depot and return to it after finish trip can be considered as DARP. In contrast to the single depot of DARP, a carpool participant in CSP is free to join in a role of a driver or a passenger and he/she can have their own departure and destination.

A GA-based approach is proposed to solve the CSP in respect to optimization. The processing time is reduced by the evolution termination criteria of early stop [4]. In addition to the GA-based method [5], a fuzzy logic controller that considers genetic diversity controls crossover and mutation probabilities during the evolution process. Jiau and Huang [6] developed a compact GA approach to reduce memory requirements by using probabilistic model to simulate the genetic population. More recently, research attention has been focused on applications that implement ridesharing services, such as Lyft Carpool,[1] UberPool,[2] BlueNet,[3] and so on. In these transportation applications, it is highly important to design effective algorithmic methods to not only to find suitable matching results for drivers and passengers but also provide drivers and passengers satisfactory route planning results.

To address the fundamentals of the CSP in this paper, two of the most important objective problems are taken into consideration. These are: 1) *Matching:* maximizing the total

Manuscript received January 29, 2018; revised April 5, 2018; accepted May 25, 2018. This work was supported by the Ministry of Science and Technology, Taiwan, under Grant 105-2923-E-027-001-MY3, Grant MOST 106-2221-E-027-017-MY3, and Grant MOST 106-2221-E-027-126-MY2. This paper was recommended by Associate Editor H. Ishibuchi. *(Corresponding author: Shih-Chia Huang.)*

The authors are with the Department of Electronic Engineering, National Taipei University of Technology, Taipei 10608, Taiwan (e-mail: schuang@ntut.edu.tw).

Digital Object Identifier 10.1109/TCYB.2018.2844324

[1]Lyft: https://www.lyft.com/.
[2]Uber: https://www.uber.com/.
[3]BlueNet: https://www.bluenet-ride.com/.

amount of users matched and 2) *Distance Cost:* minimizing distance-related metrics for drivers and passengers. These functions aim to solve the combination problem of the CSP, but often present competing and conflicting objectives. Previous research into the CSP [1], [4], [6] has shown that it is generally possible to combine multiple objectives into a single objective through a weighted linear and/or nonlinear combination of different objectives, and then solve it via any single objective optimization algorithm. The weighted approach is dependent on the choice of the relative weights assigned to the different objectives [4]. Usually, the nonlinear approach is able to avoid making the difficult choice of the relative weights by using a nonlinear function to prioritize objective importance [6].

However, while combining different objectives into a single objective ensures that the single objective function is optimized, there is no guarantee of the performance qualities of the respective objectives as they are not considered separately. Instead of compiling multiple objectives into single objectives, the multiple objectives should be optimized simultaneously and fairly while determining the solution to the CSP. This is so-called multiobjective optimization problem (MOP). Evolutionary algorithms (EAs) can be effective for resolving MOPs, and when used as such are called multiobjective EAs (MOEAs).

The presence of multiple objectives in an MOP usually gives rise to a set of tradeoff optimal solutions, also known as Pareto-optimal solutions. The set of all Pareto-optimal solutions is called the Pareto front (PF). Since the number of Pareto-optimal solutions is extensive in CSPs, it is crucial to find the best PF in an efficient way. A number of different EAs have been proposed to solve MOPs by endeavoring to find a solution set that is approximate to PF [7]–[9]. One of these MOEAs is non-dominated sorting GA II (NSGA-II). NSGA-II is well suited for solving MOP and many of previous researches have demonstrated its efficiency for MOPs [10]–[14]. In an EA (e.g., GA), a population of candidate individuals that are solutions to an optimization problem is evolved toward better solutions. Evolving populations to the next generation relies on evolutionary operators such as crossover and mutation. Moreover, the representation of a population has a significant influence on the evolutionary operator and the quality of a solution. A binary coded crossover (BX) operator is considered a simple and powerful evolutionary GA operation. In the binary-coded domain, the driver-passenger matching representation value is 1 if the passenger is matched with the driver; otherwise it is 0. However, the BX operator excessively conducts elements (passengers) to 0 values. Such a sparse matrix will result in inactive evolution for the population.

The set-based concept operated by arithmetic operators has been proved to be promising in particle swarm optimization (PSO). Chen *et al.* [15] improved representation by a set-based scheme and introduced the probability theory into enabling set-based PSO (S-PSO) method solve for some combinatorial optimization problem in discrete space. Gong *et al.* [16] proposed S-PSO to solve the discrete combinatorial problem vehicle routing problem with time windows, the route number and total distance are considered been combined into a weighted linear aggregation function.

The simulated binary crossover (SBX) was proposed for the real-coded domain and it shows a good search ability on a continuous problem [17] but not a discrete problem. Motivated by the success of SBX, we adopt both probability and set theory into the SBX algorithm to develop an advanced crossover approach called set-based SBX (SSBX) so that the SSBX evolves the population over the discrete domain.

In this paper, we develop a set-based simulated binary and multiobjective carpool matching algorithm (SSB-MOCM) to solve multiobjective optimization CSP (MOCSP) in the form of a set-based representation. The rest of this paper is divided into Sections II–IV. Our proposed SSB-MOCM is described in Section II. Section III presents the experimental results of addressing MOCSP through our proposed SSB-MOCM algorithm. Finally, our conclusions are given in Section IV.

II. Proposed Method

The NSGA has been proven efficient for solving MOPs [18]. In this section, we present an effective NSGA-based approach called the SSB-MOCM for resolving the MOCSP. The proposed SSB-MOCM algorithm is divided into two modules: 1) the set-based evolution initialization (SBEI) module and 2) the set-based multiobjective optimization (SBMOO) module. Initially, the proposed SBEI module designs effective set-based representation and population initialization for the multiobjective optimization of CSP. After the SBEI module is accomplished through effective representation and initialization, the next SBMOO module can execute its operation more efficiently. We improved the SBX by ensuring the binary crossover (BX) properties have an evolution power similar to BX, and introducing the probability theory into an SSBX for the operating set. Subsequently, the proposed SBMOO module can determine the optimum solutions, which are closer to the PF.

A. Set-Based Evolution Initialization

1) Set-Based Individual Representation: Finding the best driver and passenger matches via a GA evolution approach depends on an effective individual representation that facilitates the evolutionary operations efficiently. In the classic GA, discrete objective optimization problems are usually solved by encoding the individual as a binary string [19]. Unfortunately, in the case of CSP, generating a population during a binary representation of individuals can result in either premature convergence or inactive evolution. In contrast to the adoption of binary representation of the population in NSGA [18], our proposed set-based population representation can more effectively determine superior routes and driver/passenger matching results via evolution.

An instance of string binary representation of matching results M_j^i in CSP is as follows:

$$M_j^i = \left\{ m_j^i \mid i \in \text{driver and } j \in \text{passenger} \right\} \tag{1}$$

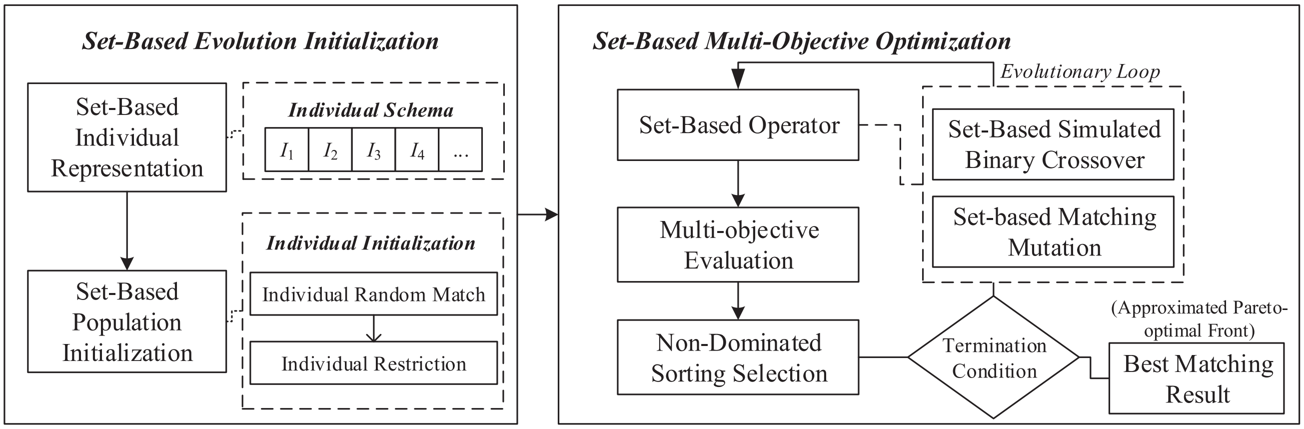

Fig. 1. Flowchart of the proposed SSB-MOCM algorithm.

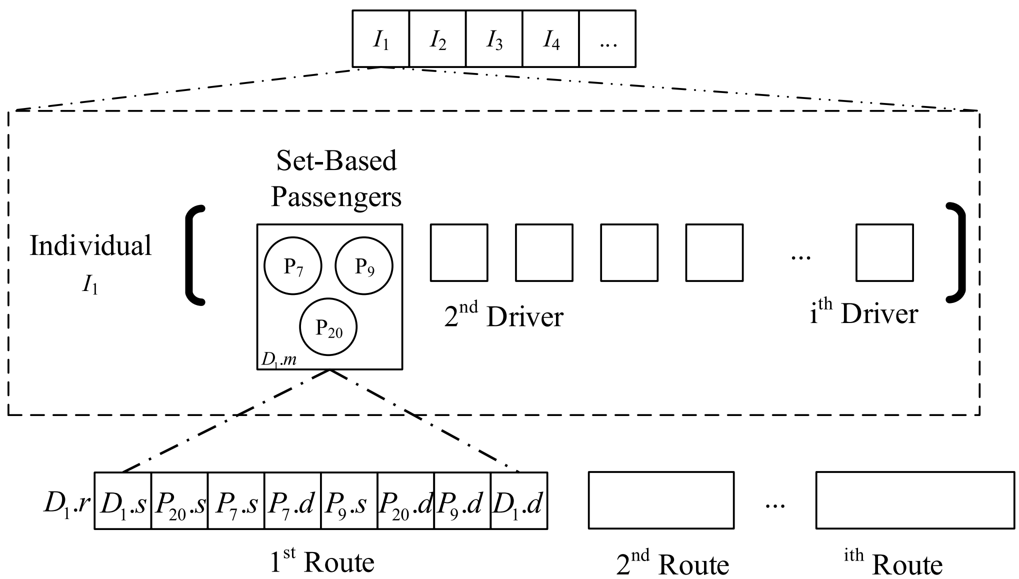

Fig. 2. Set-based-coded individual representation.

where m_j^i is the matching relationship between driver i and passenger j. m_j^i is labeled as 1 to signify that driver i and passenger j are matched; otherwise, it is labeled as 0.

However, the driver and passenger matching results are encoded as a string binary representation that incurs an inefficient crossover operation. For example, we have five drivers and ten passengers in the matching pool. If the first driver matches with the second passenger, the corresponding binary string of matching results is encoded as $\{0, 1, 0, 0, 0, 0, 0, 0, 0, 0\}$. If the second driver matches with the first and the fifth passengers, the corresponding binary string of the matching result is $\{1, 0, 0, 0, 1, 0, 0, 0, 0, 0\}$.

The sparse binary representation of the driver and passenger matching results usually suffers from stagnant crossover results. This is particularly true when the number of drivers and passengers increases, culminating in a sparser corresponding binary representation of matching results. The use of set-based individual representation becomes essential for evolution to prevent premature convergence.

In this section, we present a set-based-coded individual representation for CSP to facilitate the crossover operation.

In set-based-coded representation, each individual is formed by $I = \{D_i \mid \forall i \in D\}$ where D_i is ith of drivers, and D is the total set of drivers. A passenger group is formed by $P = \{P_j \mid \forall j \in P\}$, where P_j is jth of the passengers, and P is the total set of passengers. D_i and P_j have different properties, respectively. In Fig. 2 [20], which illustrates the set-based-coded individual.

Properties are described as follows.

1) D_i*.DID:* The unique identifier of the driver.
2) D_i*.s:* The departure location of the driver, which is represented as $\{x_0, y_0\}$.
3) D_i*.d:* The destination location of the driver, which is represented as $\{x_1, y_1\}$.
4) D_i*.c:* The capacity of the drivers vehicle.
5) D_i*.c.m:* The current matched passenger's required seat number.
6) D_i*.DIS:* The distance between the departure and destination locations of the driver.

7) *D_i.RDIS:* The distance of the ride-share. (In other words, the distance of $D_i.r$.)
8) *D_i.m:* The passengers who matched with the driver D_i. For example, the matched passengers are P_3, P_7, and P_{20}.
9) *D_i.r:* The current route of the driver, which is a set of passenger and driver routes. For example, if passengers P_7, P_9, and P_{20} are matched with driver D_1, the route could be $D_1.s \rightarrow P_{20}.s \rightarrow P_7.s \rightarrow P_7.d \rightarrow P_9.s \rightarrow P_{20}.d \rightarrow P_9.d \rightarrow D_1.d$.
10) *P_j.PID:* The unique identifier of the passenger.
11) *P_j.s:* The departure location of the passenger, which is represented as $\{x_0, y_0\}$.
12) *P_j.d:* The destination location of the passenger, which is represented as $\{x_1, y_1\}$.
13) *P_j.nec:* The number of seats required by the passenger.
14) *P_j.DIS:* The distance of the departure and destination locations of the passenger.
15) *P_j.RDIS:* The distance of the riding-share.
16) *I.fitness:* The objective functions of the individual represented as $\{f_1, f_2, f_3\}$.

2) Set-Based Population Initialization: In EAs, population initialization is an important task that affects the convergence speed and the quality of the final population [21]. Each driver can more efficiently use the vehicle and reduce the total number of vehicles on the roads by serving one or more passengers. The passengers are both evenly and fairly distributed to drivers so no driver is responsible for excessive passengers, which is a root cause of driver routing detours. To this end, we propose a restriction approach—driver detour restriction (DDR)—which restricts detours of a driver's route that would result from serving more passengers.

Our proposed initialization procedures are as follows.

1) *Step 1:* We randomly generate a parent population of size N for GA using the individual representation as discussed above:
$$\begin{aligned} \text{pop} &= \{I_1, I_2, \ldots, I_n\}, n = N \\ I_n &= \{D_1, D_2, \ldots, D_i\} \end{aligned} \tag{2}$$
where pop is the population of GA, the size of the population is N, I_n is the nth individual, each individual consist of all drivers, and D_i is the ith driver of individual.
2) *Step 2:* A free driver D_{free} is randomly selected from the individual
$$D_{\text{free}} = D_r \in D \tag{3}$$
where D_{free} is randomly selected from the total set of drivers D in the individual, and r is a random number generated in $[1, i]$ and i is the size of the total set of drivers D.
3) *Step 3:* A free passenger P_{free} is randomly selected from the total set of passengers
$$P_{\text{free}} = P_r \in P \tag{4}$$
where P_{free} is randomly selected from the total set of passengers P, and r is a random number generated in $[1, j]$ and j is the size of the total set of passengers P.
4) *Step 4:* Generate the route in respect to the driver and passenger matching results. The following are the steps for generating the route in respect to the matching results.
   a) *Step 4.1:* Calculate the number of seats required by P_{free}. If D_{free} can satisfy the seat requirement of P_{free}, then find all routes for P_{free} in $D_{\text{free}}.r$. Return to step 3 if no route is available or seat requirement is more than provided by the driver.
   b) *Step 4.2:* The matched passenger is inserted into the all routes found in step 4.1, and all routes are candidates (R_c) for matching results.
   c) *Step 4.3:* In order to compare the routes between R_c and the driver's original route, we need to calculate the distance of candidates.
   d) *Step 4.4:* The proposed approach DDR is used to determine the distance ratio between the original route and matched passenger routes to determine whether R_c is accepted or rejected
$$\begin{aligned} &\text{Matching Result} \\ &= \begin{cases} \text{accept}, & \text{if } \frac{\text{MD}-D_{\text{free}}.\text{dis}}{\text{MD}} < \text{DDRR} \\ \text{reject}, & \text{otherwise} \end{cases} \end{aligned} \tag{5}$$
   where D_{free}.dis is the distance of the departure and destination locations of the driver and MD is the distance of the candidate. Acceptance of R_c is defined as the ratio of detour, which must be lower than the DDR ratio (DDRR); otherwise, the R_c is rejected.
   e) *Step 4.5:* The matching route result is selected from the R_c, which is the smallest distance and determined during step 4.4.
   f) *Step 4.6:* If one of R_c is accepted, the passenger P_{free} is sampling without replacement.
5) *Step 5:* Repeat step 3 until all passengers are selected at least once or there is no more seat for P_{free}.
6) *Step 6:* Return to step 2 and repeat the steps until all passengers or all drivers are selected at least once in steps 2 and 3.

B. Set-Based Multiobjective Optimization

1) Set-Based Operator:

a) Set-based simulated binary crossover: In the multiobjective GA, a crossover is a stochastic search operator used to generate an offspring individual from one generation to next. Previous research has shown the BX, which is used to generate individuals, is efficient for the evolutionary strategy of binary-coded GAs. Consequently, SBX was proposed to simulate the operation of BX, which is based on an observed probability distribution applied to a real-coded domain. Moreover, SBX was designed in respect to single-point BX properties [17]. Two of these properties are the average property and the spread factor property.

Average Property: Fig. 3 shows that the average value of two arbitrary real-coded parents (P) is equal to the

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

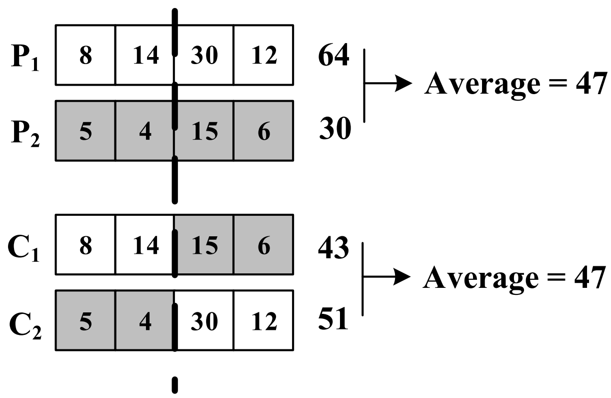

Fig. 3. Average property. (a) Average value of two arbitrary real-coded parents. (b) Average value of their offspring after the crossover operation. Note that the average values of real-coded parents *P* and their offspring *C* are equal.

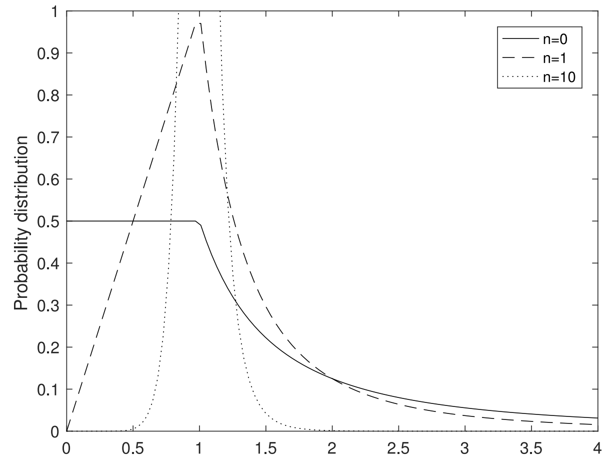

Fig. 4. Spread factor property. The probability distribution of β to simulate the result of single-point crossover in binary-coded GAs.

average value of their offspring (C). Thus, the SBX equations are as follows:

$$c_1 = \bar{x} - \frac{1}{2} \cdot \beta \cdot (p_2 - p_1) \tag{6}$$

$$c_2 = \bar{x} + \frac{1}{2} \cdot \beta \cdot (p_2 - p_1) \tag{7}$$

where p_1 and p_2 are parent individuals, c_1 and c_2 are two new offspring individuals produced by mating two parent individuals, $x = (1/2) \cdot (p_2 + p_1)$ is the average value of the parent, and β is the spread factor. As a result, the average property ensures that the offspring is similar to the parent, and that parents with good genes can pass those gene to the offspring.

Spread Factor Property: As given in the equations above, β is a spread factor that represents a ratio difference between parent individuals and offspring individuals. There are three different β probability distributions: 1) contracting crossover in which the offspring encloses the parent if $\beta > 1$; 2) expanding crossover in which the offspring is enclosed by the parent if $\beta < 1$; and 3) a situation in which the offspring is the same as the parent if $\beta = 1$. As such, the probability distribution of β in SBX was designed to approximate the probability distribution of β during binary-coded crossover. The SBX spread factor is shown as follows:

$$\mathrm{c}(\beta) = \begin{cases} 0.5(n+1) \cdot \beta^n, & \text{if } \beta \le 1 \\ 0.5(n+1) \cdot \frac{1}{\beta^{n+2}}, & \text{if } \beta > 1 \end{cases} \tag{8}$$

where n is commonly between 2 and 5, with larger values indicating closer simulation of single-point binary-coded crossover. This allows the offspring to be mutated by the spread factor, resulting in superior offspring.

Unfortunately, real-coded GAs with the SBX operator perform well only for specific optimization problems that feature continuous search space [17]. Because of this, the real-coded SBX operator is not applicable for set-based individuals in the CSP.

In this section, we propose a new crossover operator called SSBX, which is based on SBX and is introduced through both probability and set theory. SSBX inherits two characteristic properties from SBX: 1) average property and 2) spread factor property. The set theory is adopted to redefine the mathematical operator. In SSBX, the real number of addition $(+)$ and subtraction $(-)$ operations are redefined as the union set $(\oplus)$ and complement set $(\ominus)$, respectively, as

$$c_1[i] \triangleq \bar{x}[i] \ominus x^{-}_{0.5\cdot\beta}[i] \tag{9}$$

$$c_2[i] \triangleq \bar{x}[i] \oplus x^{-}_{0.5\cdot\beta}[i] \tag{10}$$

where i is the index of the element in the set, and the $\bar{x}[i]$ and $x^{-}_{0.5\cdot\beta}[i]$ are as follows:

$$\bar{x}[i] \triangleq \frac{1}{2}(x_1[i] \oplus x_2[i]) \tag{11}$$

$$x^{-}_{0.5\cdot\beta}[i] \triangleq \frac{1}{2} \cdot \beta \cdot (x_2[i] \ominus x_1[i]) \tag{12}$$

where x_1 and x_2 are the parents of the individual, and β is the probability factor. In Fig. 5(a) of the set operation, the spread factor is the probability that each element is retained. To operate with the set, we assign a probability for each element.

In (13) and (14), each element in $\bar{x}$[i] is assigned a value of 0.5, and each element in $x^{-}_{0.5\cdot\beta}$[i] is assigned a factor $0.5 \cdot \beta$ as shown in Fig. 5(b)

$$\begin{aligned} \bar{x}[\mathrm{i}] &\triangleq \frac{1}{2}(x_1[i] + x_2[i]) \\ &= \left\{ \frac{e}{0.5} \mid e \in x_1[i] \cup x_2[i] \right\} \end{aligned} \tag{13}$$

$$\begin{aligned} x^{-}_{0.5\cdot\beta}[\mathrm{i}] &\triangleq \frac{1}{2} \cdot \beta \cdot (x_2[i] - x_1[i]) \\ &= \left\{ \frac{e}{0.5 \cdot \beta} \mid e \in x_2[i] \text{ and } e \notin x_1[i] \right\}. \end{aligned} \tag{14}$$

Inspired by (8), the defined element probability distribution β can be any point under the curve, as shown in Fig. 4. As shown in Fig. 5(c), an average probability distribution β is produced through a random value **rnd** in the range [0, 1], using the integral of (8)

$$\mathrm{c}(\beta) = \begin{cases} (2 \cdot \mathrm{rnd})^{\frac{1}{n+1}}, & \text{if rnd} \le 0.5 \\ \left(\frac{1}{2\cdot(1-\mathrm{rnd})}\right)^{\frac{1}{n+1}}, & \text{if rnd} > 0.5 \end{cases} \tag{15}$$

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

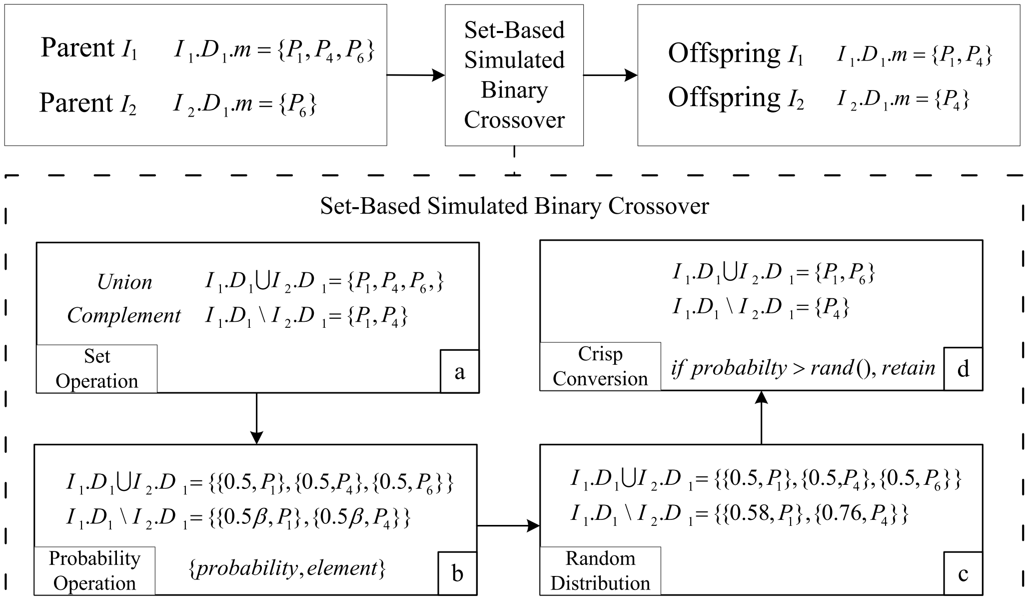

Fig. 5. Set-based simulated binary crossover flowchart.

where n can control the probability distribution, which is usually between 2 and 5. In Fig. 5(d), the crossover operation is defined via the corresponding equation (16), (17) as follows:

$$\begin{aligned} c_1[i] &\triangleq \bar{x}[i] \ominus x^{-}_{0.5\cdot\beta}[i] \\ &= \left\{ e \mid e \in \mathrm{crisp}(\bar{x}[i]) \text{ and } e \notin \mathrm{crisp}(x^{-}_{0.5\cdot\beta}[i]) \right\} \end{aligned} \tag{16}$$

$$\begin{aligned} c_2[i] &\triangleq \bar{x}[i] \oplus x^{-}_{0.5\cdot\beta}[i] \\ &= \left\{ e \mid e \in \mathrm{crisp}(\bar{x}[i]) \ \cup \ e \in \mathrm{crisp}(x^{-}_{0.5\cdot\beta}[i]) \right\} \end{aligned} \tag{17}$$

where crisp$(\cdot)$ conversion determines whether each element is retained by a random number, with the higher probability values of the element indicating a higher probability that the element will be retained. As a result, two new progeny solutions are generated by union set ($\oplus$) or complement set ($\ominus$) of the crisp$(\cdot)$ function transformation results.

b) Set-based matching mutation: Subsequently, the set-based matching mutation is applied to the new offspring. The procedure of the matching mutation is shown in Algorithm 1. The purpose of this mutation is to find an available driver for passengers. In other words, the total number of matched passenger (f_1) can be optimized. However, the available passenger (A) is the difference between P and M. To enrich the diversity of offspring, RandomOrder() is used to shuffle the order of driver (D). The GenerateRoute() search available route to the driver for passenger complies with the **DDR** mentioned in Section II-A.

2) Multiobjective Evaluation: In this section, two of the objective problems for MOCSP are taken into consideration for evaluating the passenger-to-driver matching results: 1) matching, in which the total number of matched passengers is maximized and 2) distance cost, in which the additional ride distance accrued by matching passengers to drivers is minimized.

Algorithm 1: Matching Mutation

```
Input:
  P, a set of passengers.
  M, a set of passengers matched with driver.
  D, a set of drivers in each new offsprings.
begin
    A ← P − M
    RandomOrder(D)
    foreach d ∈ D do
        foreach p ∈ A do
            if d.c ≥ p.nec then
                R_new ← GenerateRoute(d, p)
                if R_new ≠ empty then
                    d.m ← d.m + p
                    d.r ← R_new
                    A ← A − p
```

The objective functions are given as follows.

1) f_1*:* Total number of matched passengers

$$\max \quad f_1 = |M| = \sum_{i \in D} (|D_i.m|)$$

where $D_i.m$ is a set of passengers matched with a driver, and D is the total set of drivers.

2) f_2*:* Travel route distance of drivers

$$\min \quad f_2 = \sum_{i \in D} \sum_{p,q \in D_i.r} d(p, q)$$

where $D_i.r$ is a set of route points of a driver, p and q are the points in the set, and D is the total set of drivers. $d(\cdot)$ is a distance evaluation such as straight-line distance between two points in Euclidean space.

3) f_3*:* Average travel distance of matched passengers

$$\min = f_3 = \frac{\sum_{j \in P} P_j.\mathrm{RDIS}}{|m|}$$

where P_j.RDIS is the ride-sharing distance of the passenger, m is the total number of matched passengers, and P is the total set of passengers. As given above, the objective functions are assigned to I.fitness.

To produce a valid and feasible individual via the crossover operation, it is necessary to comply with the constraints as follows.

1) *Dynamic Capacity:* The capacity of the drivers vehicle must provide enough seats for passengers. The behavior of a passenger entering a vehicle will consume the seat capacity of $D_i.c$. On the contrary, the behavior of a passenger exiting a vehicle will restore the seat capacity of $D_i.c$

$$D_i.c.m \leq D_i.c, \ \ \forall i \in D \tag{18}$$

where $D_i.c.m$ represents the total accumulation of matched passengers' required seats, and $D_i.c$ is the capacity of the driver's vehicle.

2) *Single Service Constraint:* In any matching result, all passengers are served by at most one driver as defined in the S-to-M carpooling arrangement [6], [22]

$$\{k \ |k \in M \text{ and } k = j\}| \leq 1 \, , \forall j \in P \tag{19}$$

where M indicates a set of passengers matched with any driver. k indicates matched counting for each passenger in the matching result, if k is large than 1 the passenger is removed from the matched driver.

3) *Pick-Up and Drop-Off Constraint:* The driver must first travel to the departure location of a passenger. Then, the driver can travel to the destination location of the passenger

$$s_k(D_i.r) < d_k(D_i.r), \ \ \forall k \in D \cup P, \ \ \forall i \in D \tag{20}$$

where $s_k(D_i.r)$ returns the index of $P_j.s$ in $D_i.r$ if $P_j.s$ exists in $D_i.r$; otherwise the value is 0. And $d_k(D_i.r)$ returns the index of $P_j.d$ in $D_i.r$ if $P_j.d$ exists in $D_i.r$; otherwise, the value is 1.

4) *Route Completion Constraint:* For a passenger, a pair of departure and destination locations must be completely traversed by the driver matched with the passenger

$$|O_k| = |E_k|, \ \ \forall k \in D \cup P, \ \ \forall i \in D \tag{21}$$

where $O_k = \{n \ |n \in D_i.r \text{ and } n = s_k\}$ attempts to select node s_k from route $D_i.r$, and, similarly, $E_k = \{n \ |n \in D_i.r \text{ and } n = d_k\}$ also attempts to select node d_k from route $D_i.r$.

3) Nondominated Sorting Selection: Through the operations discussed in the crossover section, the size of the population R_t is doubled. The population consists of individuals I_t, where t is the number of generation, and is divided into two groups of individuals P_t and Q_t that correspond to parents and offspring, respectively. The size of R_t is $2N$ since the size of P_t and Q_t are equal to N. R_t are then ranked by the fast nondominated sorting approach [18] (all objective functions of f_1–f_3 are considered based on the Pareto dominance relation). After solution were ranked, the crowding distance, is applied to select solution in larger crowding distance to ensure the diversity of solution. Finally, N solutions are selected from the R_t as new parent for next evolutionary loop or the final approximated PF if the evolutionary loop is done.

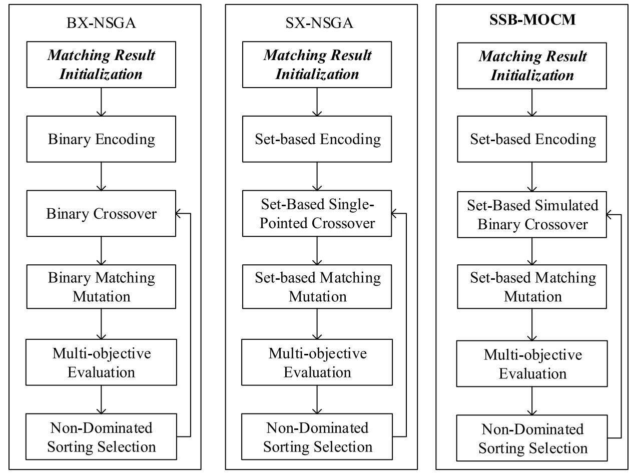

Fig. 6. Models of all compared algorithms. BX-NSGA evolve the population in binary domain. In contrast, SX-NSGA and proposed SSB-MOCM are operating in set-based domain.

III. Experimental Results

In this section, the proposed SSB-MOCM is compared with binary-coded single-pointed crossover (BX) based on NSGA (BX-NSGA), and set-based single-pointed crossover (SX) based on NSGA (SX-NSGA). The goal of MOEAs is to find a good solution set that approximates PF [23]–[25]. But when it comes to a comparison of the different optimal solution sets produced by these different optimization approaches, it can be difficult to intuitively quantitatively evaluate the efficacy of each approach. However, we were able to evaluate the performance of different MOEAs by using the same test instances for each approach, along with a suite of various performance metrics.

The model of all compared algorithm is shown in Fig. 6. All of the compared methods in this paper are tested with the same CSP instances: synthetic metropolis-inspired datasets that embody three sets of spatial distributions of commuting patterns and motion in a city center [26], [27]. The performance of BX-NSGA, SX-NSGA, and SSB-MOCM algorithm were evaluated by four different performance metrics: 1) C-metric (C); 2) spacing metric (SP); 3) inverted generational distance (IGD); and 4) hypervolume (HV).

Three different moving patterns were used in the analysis: 1) inward radiating movement (CI); 2) lateral drifting movement (CL); and 3) outward radiating movement (CO) [4]. Factors were varied, including the number of input drivers, the number of input passengers, the number of seats provided by the driver, the number of seats required by the passenger. The range of passenger and driving seat is treated as a four-seater car. In total, datasets reflecting 24 different instances were

TABLE I
QUANTITATIVE RESULTS OF TOTAL 24 DIFFERENT INSTANCES FOR BX-NSGA, SX-NSGA, AND THE PROPOSED SSB-MOCM ALGORITHM VIA C-METRIC, IGD, SP, AND HV

Type	C-Metric		IGD			SP			HV		
	C(SSB-MOCM,BX-NSGA)	C(SSB-MOCM,SX-NSGA)	BX-NSGA	SX-NSGA	SSB-MOCM	BX-NSGA	SX-NSGA	SSB-MOCM	BX-NSGA	SX-NSGA	SSB-MOCM
CI-15-40	0.9947	0.9675	0.0850	0.1395	0.0223	0.0125	0.0210	0.0097	2.2807	2.3970	2.6378
CI-20-30	1.0000	0.9850	0.0746	0.1295	0.0087	0.0095	0.0246	0.0067	1.9924	2.0769	2.3352
CI-20-40	1.0000	0.7463	0.1570	0.0679	0.1265	0.0101	0.0218	0.0084	2.2891	2.5080	2.5129
CI-30-45	1.0000	0.9850	0.1046	0.1018	0.0308	0.0069	0.0179	0.0061	1.8374	1.9364	2.1449
CI-30-60	1.0000	0.9675	0.0782	0.1114	0.0257	0.0069	0.0163	0.0063	2.1345	2.2617	2.4191
CI-45-65	1.0000	1.0000	0.0676	0.1084	0.0000	0.0065	0.0328	0.0050	1.6910	1.7131	1.9539
CI-45-90	1.0000	0.9724	0.0610	0.0740	0.0180	0.0053	0.0188	0.0047	1.9451	2.0561	2.1767
CI-60-90	1.0000	0.9387	0.1388	0.0604	0.0873	0.0044	0.0218	0.0045	1.6342	1.7743	1.8785
CL-15-40	0.9948	0.8788	0.1002	0.1069	0.0569	0.0116	0.0235	0.0117	2.2329	2.3303	2.5528
CL-20-30	1.0000	1.0000	0.0487	0.1027	0.0000	0.0102	0.0265	0.0061	2.0740	2.1088	2.3434
CL-20-40	0.9950	0.8888	0.0992	0.1093	0.0561	0.0117	0.0223	0.0116	2.2963	2.4324	2.5973
CL-30-45	0.9907	0.9600	0.0617	0.1090	0.0254	0.0110	0.0182	0.0083	2.2986	2.2875	2.5191
CL-30-60	0.9503	0.8538	0.1068	0.0940	0.0783	0.0077	0.0179	0.0081	2.1914	2.3487	2.4546
CL-45-65	1.0000	0.9363	0.0787	0.1067	0.0407	0.0071	0.0176	0.0068	2.2303	2.3683	2.4865
CL-45-90	0.8961	0.8713	0.1124	0.1044	0.0823	0.0066	0.0179	0.0060	2.1588	2.2548	2.3858
CL-60-90	0.9800	0.9863	0.0549	0.0876	0.0153	0.0068	0.0169	0.0070	2.1274	2.2322	2.3536
CO-15-40	0.9725	0.6963	0.1753	0.0751	0.1495	0.0096	0.0191	0.0117	2.2272	2.4383	2.4498
CO-20-30	1.0000	0.9721	0.0818	0.1044	0.0256	0.0085	0.0239	0.0062	1.9594	2.0338	2.2507
CO-20-40	1.0000	0.9863	0.0686	0.1382	0.0097	0.0091	0.0199	0.0073	2.1926	2.2521	2.5416
CO-30-45	1.0000	0.9663	0.1271	0.0930	0.0642	0.0076	0.0233	0.0055	1.7966	1.9132	2.0998
CO-30-60	1.0000	0.9492	0.1165	0.0703	0.0624	0.0072	0.0209	0.0050	1.8620	2.0437	2.1581
CO-45-65	1.0000	1.0000	0.0655	0.0859	0.0000	0.0067	0.0187	0.0065	1.6837	1.7257	1.9361
CO-45-90	1.0000	0.8536	0.2148	0.0542	0.1820	0.0061	0.0255	0.0044	1.8140	1.9614	2.0574
CO-60-90	1.0000	1.0000	0.0636	0.0700	0.0000	0.0046	0.0347	0.0045	1.6490	1.7532	1.8977
Mean	0.9906	0.9317	0.0976	0.0960	0.0487	0.0081	0.0217	0.0070	2.0249	2.1337	2.2976
σ	0.0228	0.0787	0.0412	0.0231	0.0491	0.0023	0.0047	0.0023	0.2237	0.2391	0.2309

prepared. The remaining test configurations for BX-NSGA, SX-NSGA, and SSB-MOCM algorithm are: population size = 40, and iteration = 1000. The three compared algorithms were implemented in Python 2.7. Each test was conducted twenty times independently.

A. Performance Metrics in Multiobjective Optimization

An MOEA consists of two goals for obtained solutions: 1) effective convergence to the PF and 2) effective diversity. In addition, there are three major performance criteria of an MOEA: 1) capacity; 2) convergence; and 3) diversity [28]. As such, we adopt four performance metrics based on these goals and criteria for three evaluated objectives of MOCSP in this paper. In order to evaluate the capacity metrics, the coverage of two sets (C-metric) is adopted to determine the ratio of nondominated solutions between two optimal solution sets. Moreover, the SP supplies the distribution and spread of solution sets, and determines the diversity of the obtained solution sets. Furthermore, two convergence-diversity (CD) metrics are used: IGD and HV. The CD metrics determine if both convergence and diversity of the optimal solution set are on a single scale.

The definition of four performance metrics is given as follows.

1) *C-Metric (C):* Zitzler and Thiele [29] proposed a commonly used metric for comparing two sets of nondominated solutions, S_1 and S_2. The C-metric can quantify the coverage of two sets by using the percentage of the solutions in S_2 that are dominated by at least one solution in S_1. We denoted the metric as $C(S_1, S_2)$, which is defined as the percentage of the solutions in S_2 that are dominated by at least one solution in S_1. If $C(S_1, S_2) = 1$, then all nondominated solutions in S_2 are dominated by solutions in S_1. If $C(S_1, S_2) = 0$, then all nondominated solutions in S_1 are dominated by solutions in S_2. The sum of $C(S_1, S_2)$ and $C(S_2, S_1)$ is not always equal to 1 since some solutions in S_1 and S_2 may not dominate each other.
2) *SP:* The SP was first proposed by Schott [30]. The main idea of this metric is to calculate the relative distance between consecutive solutions and the obtained nondominated set. The mathematical expression of SP is as follows:
$$\mathrm{SP}(S) = \sqrt{\sum_{i=1}^{|S|} \frac{(d_i - \bar{d})^2}{|S| - 1}} \tag{22}$$
where $d_i = \min_{\vec{s}_j \in S, \vec{s}_j \neq \vec{s}_i} \|F(\vec{s}_i) - F(\vec{s}_j)\|$ and $s_i \in S$. A low SP value indicates a greater number of evaluated results, and a greater chance of obtaining a more equitably spread solution close to the Pareto-optimal front [31].
3) *IGD:* In [32]–[36], the metric of IGD has a similar formulation to generational distance and is used for measuring both the convergence and diversity of the nondominated solutions obtained. The mathematical formulation of IGD is given as follows [28]:
$$\mathrm{IGD}(P, S) = \frac{\left(\sum_{i=1}^{|P|} {d_i}^q\right)^{\frac{1}{q}}}{|P|} \tag{23}$$

TABLE II
AVERAGE OF DISTANCE OBJECTIVE f_2 AND f_3 OF ALL COMPARED METHODS OF MOCSP

Algorithm	f_2	f_3
BX-NSGA	18116.46	13041.41
SX-NSGA	17621.65	28524.94
SSB-MOCM	16775.25	11766.54

where $d_i = \min s_{s_i \in S} \| F(\overrightarrow{p_i} - F(\overrightarrow{s})) \|$, $\overrightarrow{p_i} \in P, q = 2$, and d_i is the smallest distance of $\overrightarrow{p_i} \in P$ to the closest solutions in S.

4) *HV:* This metric is proposed by Zitzler *et al.* [29], [37]–[39]. The mathematical formulation of HV is given as follows [28]:

$$\mathrm{HV}(S, R) = \mathrm{volume}\left(\bigcup_{i=1}^{|S|} v_i\right) \tag{24}$$

This gives the volume (in the objective space) that is dominated by the optimal solution set S and R is the reference set. A larger HV indicates that the corresponding nondominated solutions are closer to PF [40].

To evaluate the efficacy of the passenger-to-driver matching results of each compared EA in regard to MOCSP, the Pareto-optimal Front needs to be compared to the approximate PF solutions. Generating the Pareto-optimal solutions front consists of three generative steps: 1) combine all solutions obtained by three compared algorithms as a set; 2) apply nondominated rank sorting to the set; and 3) regard the first rank level of nondominated solutions as the optimal PF [18].

B. Quantitative Comparison of BX-NSGA, SX-NSGA, and Proposed SSB-MOCM Algorithm

The quantitative results are shown in Table I. Four metrics—C-metric, IGD, SP, and HV—take three objective functions into consideration for three different types benchmarks: 1) CI; 2) CL; and 3) CO. The extreme values of three objective function of MOCSP were recorded from the mixed population of all compared algorithms, a total of 120 solutions. The objectives of solutions were normalized in the range of [0, 1]. And the reference points of HV metric were defined as [1.5, 1.5, 1.5] to calculate the HV of each solution of the compared algorithm. It can be observed that the SSB-MOCM yields IGD, SP, and HV values that are consistently superior to those produced by the other compared methods. Moreover, the field of C-metric shows the coverage of solution set obtained by two compared algorithm. SSB-MOCM perform the best results of all three different moving pattern. Especially, C(SSB-MOCM, BX-NSGA) is close to 1 in most of the test instances. It proved that binary-coded representation is not able to provide better solutions set against to set-based representation. In addition, the average of distance objective function f_2 and f_3 is shown in Table II. To calculate the average of the distance objective of all compared instance. In each test instance, we select one solution which achieved maximum matched number of passenger (objective function f_1) and lowest f_2 and one solution achieved maximum f_1 and lowest f_3 for each compared algorithm, total two solutions is selected. Two solutions have the best optimal result of objective f_2 and f_3, respectively. This ensures the distance comparison is fair and reasonable. The proposed SSB-MOCM achieved the lowest value of f_2 and f_3 outperforms the compare algorithms.

TABLE III
KRUSKAL-WALLIS H SIGNIFICANCE TEST FOR ALL COMPARED METHODS OF PERFORMANCE METRICS OF MOCSP

Comparison	H-statistic	p-value	α
IGD	19.5708	0.00006	0.01
SP	48.9354	0.00001	0.01
HV	13.1475	0.00140	0.01

TABLE IV
MANN–WHITNEY U TEST FOR SHOWING SIGNIFICANCE BETWEEN SSB-MOCM AND OTHER METHODS FOR PERFORMANCE METRICS OF MOCSP

Comparison, IGD		U-value*	U-value	Z-score	p-value	α
SSB-MOCM* vs.	BX-NSGA	465	111	-3.63	0.00014	0.01
	SX-NSGA	479	97	3.92804	0.00004	0.01
Comparison, SP		U-value*	U-value	Z-score	p-value	α
SSB-MOCM* vs.	BX-NSGA	379.5	196.5	1.8763	0.03005	0.05
	SX-NSGA	576	0	5.92815	0.00001	0.01
Comparison, HV		U-value*	U-value	Z-score	p-value	α
SSB-MOCM* vs.	BX-NSGA	121	455	-3.43	0.0003	0.01
	SX-NSGA	178	398	2.25785	0.0119	0.05

C. Significance of SSB-MOCM Using Nonparametric Statistical Tests

In this section, the performance metrics including IGD, SP, and HV are discussed derived through statistical tests including Kruskal-Wallis H test and Mann-Whitney U test, to verify the significance of the performance of proposed SSB-MOCM using the same datasets. In Table III summarize the Kruskal–Wallis H test result of all compared algorithms in term of IGD, SP, and HV. It shows that three algorithms have significance difference at significance levels ($\alpha = 0.01$).

Furthermore, the Mann-Whitney U test is used to find the significance of two compared algorithms and the U-value, and significance level (p-value and α) are shown in Table IV. The U-value of Mann-Whitney U test shows the rank sum of larger data in comparison. In other words, the metrics expected to maximize (e.g., HV metric) the lower U-value indicates that proposed is superior; in contrast, the metrics expected to minimize (e.g., SP and IGD) larger U-value indicates the proposed is superior. As already indicated, SSB-MOCM is superior to BX-NSGA and SX-NSGA in term of SP, IGD, and HV. And the proposed method SSB-MOCM has an extremely significant level ($\alpha = 0.01$) performance advantage over BX-NSGA and SX-NSGA. As can be inferred from the comparison of capacity, convergence, and diversity via the four different metrics, the results produced by our SSB-MOCM algorithm were consistently superior to those of the other compared algorithms for all 24 different benchmarks.

IV. Conclusion

In this paper, we describe a MOCSP, which requires effective matching of drivers and passengers. The proposed SSB-MOCM algorithm successfully solves the MOCSP through two major modules: 1) SBEI and 2) SBMOO.

The SBEI module designs effective set-based representation and population initialization in MOCSP. The SBMOO module includes an advanced set-based simulated binary operation that introduces both set and probability theory to evolve the population in an efficient way. Moreover, it provides effective matching solutions that approximate the PF.

In the experimental results, a total of 24 test instances were examined through the use of our proposed SSB-MOCM and two compared approaches. Four different types of performance metrics analyzed the results of each approach and demonstrated that our proposed SSB-MOCM algorithm outperforms the other compared algorithms, resulting in superior performance in terms of capacity, convergence, and diversity.

References

[1] C.-H. Lin, M.-K. Jiau, and S.-C. Huang, "A cloud computing framework for real-time carpooling services," in *Proc. 6th Int. Conf. New Trends Inf. Sci. Service Sci. Data Min. (ISSDM)*, Taipei, Taiwan, 2012, pp. 266–271.

[2] R. M. Jorgensen, J. Larsen, and K. B. Bergvinsdottir, "Solving the dial-a-ride problem using genetic algorithms," *J. Oper. Res. Soc.*, vol. 58, no. 10, pp. 1321–1331, 2007.

[3] S. N. Parragh, J.-F. Cordeau, K. F. Doerner, and R. F. Hartl, "Models and algorithms for the heterogeneous dial-a-ride problem with driver-related constraints," *OR Spectr.*, vol. 34, no. 3, pp. 593–633, 2012.

[4] S.-C. Huang, M.-K. Jiau, and C.-H. Lin, "A genetic-algorithm-based approach to solve carpool service problems in cloud computing," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 1, pp. 352–364, Feb. 2015.

[5] S.-C. Huang, M.-K. Jiau, and C.-H. Lin, "Optimization of the carpool service problem via a fuzzy-controlled genetic algorithm," *IEEE Trans. Fuzzy Syst.*, vol. 23, no. 5, pp. 1698–1712, Oct. 2015.

[6] M.-K. Jiau and S.-C. Huang, "Services-oriented computing using the compact genetic algorithm for solving the carpool services problem," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 5, pp. 2711–2722, Oct. 2015.

[7] N. Srinivas and K. Deb, "Muiltiobjective optimization using nondominated sorting in genetic algorithms," *Evol. Comput.*, vol. 2, no. 3, pp. 221–248, Sep. 1994.

[8] C. M. Fonseca and P. J. Fleming, "Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization," in *Proc. Icga*, vol. 93, Jul. 1993, pp. 416–423.

[9] J. Horn, N. Nafpliotis, and D. E. Goldberg, "A niched Pareto genetic algorithm for multiobjective optimization," in *Proc. 1st IEEE Conf. Evol. Comput. IEEE World Congr. Comput. Intell.*, Orlando, FL, USA: IEEE, 1994, pp. 82–87.

[10] B. Milosevic and M. Begovic, "Nondominated sorting genetic algorithm for optimal phasor measurement placement," *IEEE Trans. Power Syst.*, vol. 18, no. 1, pp. 69–75, Feb. 2003.

[11] E. Zitzler, L. Thiele, and J. Bader, "On set-based multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 14, no. 1, pp. 58–79, Feb. 2010.

[12] M. B. Ercan and J. L. Goodall, "Design and implementation of a general software library for using NSGA-II with SWAT for multi-objective model calibration," *Environ. Model. Softw.*, vol. 84, pp. 112–120, Oct. 2016.

[13] D. Gong, J. Sun, and Z. Miao, "A set-based genetic algorithm for interval many-objective optimization problems," *IEEE Trans. Evol. Comput.*, vol. 22, no. 1, pp. 47–60, Feb. 2018.

[14] S.-C. Huang, M.-K. Jiau, and K.-H. Chong, "A heuristic multi-objective optimization algorithm for solving the carpool services problem featuring high-occupancy-vehicle itineraries," *IEEE Trans. Intell. Transp. Syst.*, to be published, doi: 10.1109/TITS.2017.2760017.

[15] W.-N. Chen *et al.*, "A novel set-based particle swarm optimization method for discrete optimization problems," *IEEE Trans. Evol. Comput.*, vol. 14, no. 2, pp. 278–300, Apr. 2010.

[16] Y.-J. Gong *et al.*, "Optimizing the vehicle routing problem with time windows: A discrete particle swarm optimization approach," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 42, no. 2, pp. 254–267, Mar. 2012.

[17] K. Deb and R. B. Agrawal, "Simulated binary crossover for continuous search space," *Complex Syst.*, vol. 9, no. 2, pp. 115–148, 1994.

[18] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002.

[19] X.-B. Hu and E. Di Paolo, "Binary-representation-based genetic algorithm for aircraft arrival sequencing and scheduling," *IEEE Trans. Intell. Transp. Syst.*, vol. 9, no. 2, pp. 301–310, Jun. 2008.

[20] S.-K. Chou, M.-K. Jiau, and S.-C. Huang, "Stochastic set-based particle swarm optimization based on local exploration for solving the carpool service problem," *IEEE Trans. Cybern.*, vol. 46, no. 8, pp. 1771–1783, Aug. 2016.

[21] S. Rahnamayan, H. R. Tizhoosh, and M. M. Salama, "A novel population initialization method for accelerating evolutionary algorithms," *Comput. Math. Appl.*, vol. 53, no. 10, pp. 1605–1614, May 2007.

[22] N. Agatz, A. Erera, M. Savelsbergh, and X. Wang, "Optimization for dynamic ride-sharing: A review," *Eur. J. Oper. Res.*, vol. 223, no. 2, pp. 295–303, Dec. 2012.

[23] C. A. C. Coello, "Evolutionary multi-objective optimization: A historical view of the field," *IEEE Comput. Intell. Mag.*, vol. 1, no. 1, pp. 28–36, Feb. 2006.

[24] J. J. Durillo, A. J. Nebro, and E. Alba, "The jMetal framework for multi-objective optimization: Design and architecture," in *Proc. IEEE Congr. Evol. Comput.*, vol. 5467. Barcelona, Spain: IEEE, Jul. 2010, pp. 1–8.

[25] K. Deb, *Multiobjective Optimization Using Evolutionary Algorithms*. Chichester, U.K.: Wiley, 2001.

[26] M. O. Asikhia and N. F. Nkeki, "Polycentric employment growth and the commuting behaviour in Benin metropolitan region, Nigeria," *J. Geography Geol.*, vol. 5, no. 2, pp. 1–17, Mar. 2013.

[27] B. Meng, L. Zheng, H. Yu, and G. Me, "Spatial characteristics of the residents' commuting behavior in Beijing," in *Proc. 19th Int. Conf. Geoinformat.*, Shanghai, China: IEEE, Jun. 2011, pp. 1–5.

[28] S. Jiang, Y.-S. Ong, J. Zhang, and L. Feng, "Consistencies and contradictions of performance metrics in multiobjective optimization," *IEEE Trans. Cybern.*, vol. 44, no. 12, pp. 2391–2404, Dec. 2014.

[29] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach," *IEEE Trans. Evol. Comput.*, vol. 3, no. 4, pp. 257–271, Nov. 1999.

[30] J. R. Schott, "Fault tolerant design using single and multicriteria genetic algorithm optimization," M.S. thesis, Dept. Aeronautics Astronautics, Massachusetts Inst. Technol., Cambridge, MA, USA, May 1995.

[31] S. Mirjalili and A. Lewis, "Novel performance metrics for robust multi-objective optimization algorithms," *Swarm Evol. Comput.*, vol. 21, pp. 1–23, Apr. 2015.

[32] H. Li and Q. Zhang, "Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 13, no. 2, pp. 284–302, Apr. 2009.

[33] M. R. Sierra and C. A. C. Coello, "Improving PSO-based multi-objective optimization using crowding, mutation and ϵ-dominance," in *Proc. Int. Conf. Evol. Multi Criterion Optim.*, 2005, pp. 505–519.

[34] D. A. Van Veldhuizen and G. B. Lamont, "Multiobjective evolutionary algorithm test suites," in *Proc. ACM Symp. Appl. Comput.*, San Antonio, TX, USA: ACM, 1999, pp. 351–357.

[35] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. G. da Fonseca, "Performance assessment of multiobjective optimizers: An analysis and review," *IEEE Trans. Evol. Comput.*, vol. 7, no. 2, pp. 117–132, Apr. 2003.

[36] Q. Zhang and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *IEEE Trans. Evol. Comput.*, vol. 11, no. 6, pp. 712–731, Dec. 2007.

[37] E. Zitzler and L. Thiele, "Multiobjective optimization using evolutionary algorithms—A comparative case study," in *Proc. Conf. Parallel Problem Solv. From Nat. PPSN V*, Sep. 1998, pp. 292–301.

[38] E. Zitzler, "*Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications*. Aachen, Germany: Shaker, 1999.

[39] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the strength Pareto evolutionary algorithm for multiobjective optimization," in *Evolutionary Methods for Design, Optimisation, and Control*. Barcelona, Spain: CIMNE, 2002, pp. 95–100.

[40] Y. Zhou and J. Wang, "A local search-based multiobjective optimization algorithm for multiobjective vehicle routing problem with time windows," *IEEE Syst. J.*, vol. 9, no. 3, pp. 1100–1113, Sep. 2015.

Jing-Jie Lin received the B.S. degree from the Department of Electronic Engineering, National Yunlin University of Science and Technology, Douliu, Taiwan, in 2015. He is currently pursuing the Ph.D. degree with the Department of Electronic Engineering, National Taipei University of Technology, Taipei, Taiwan.

His current research interests include artificial intelligence, optimization methods, data science, and applications of mobile application and cloud computing.

Shih-Chia Huang (SM'14) received the B.S. degree from National Taiwan Normal University, Taipei, Taiwan, the M.S. degree from National Chiao Tung University, Hsinchu, Taiwan, and the Doctorate degree in electrical engineering from National Taiwan University, Taipei, in 2009.

He is a Full Professor with the Department of Electronic Engineering, National Taipei University of Technology, Taipei, Taiwan, and an International Adjunct Professor with the Faculty of Business and Information Technology, University of Ontario Institute of Technology, Oshawa, ON, Canada. He is currently the Chair of the IEEE Taipei Section Broadcast Technology Society, and was a Review Panel Member of the Small Business Innovation Research program with the Department of Economic Development of Taipei and New Taipei City, respectively. He has published over 80 journal and conference papers and holds over 60 patents in the United States, Europe, Taiwan, and China. His current research interests include intelligent multimedia systems, image processing and video coding, video surveillance systems, cloud computing and big data analytics, artificial intelligence, and mobile applications and systems.

Dr. Huang was a recipient of the Kwoh-Ting Li Young Researcher Award by the Taipei Chapter of the Association for Computing Machinery in 2011, the 5th National Industrial Innovation Award by the Ministry of Economic Affairs, Taiwan, in 2017, the Dr. Shechtman Young Researcher Award by the National Taipei University of Technology in 2012, the Outstanding Research Award from National Taipei University of Technology in 2014, and the College of Electrical Engineering and Computer Science, National Taipei University of Technology in 2014–2016. He has been an Associate Editor of the *Journal of Artificial Intelligence* and a Guest Editor of *Information Systems Frontiers*, the *Journal of Systems Architecture*, and *Electronic Commerce Research and Applications*. He is also the Services and Applications Track Chair of the IEEE CloudCom 2016–2017 Conference, the Applications Track Chair of the IEEE BigData Congress in 2015, the General Chair of the 2015–2016 IEEE BigData Taipei Satellite Session, and the Deep learning, Ubiquitous and Toy Computing Minitrack Chair of the 2017–2018 Hawaii International Conference on System Sciences.

Ming-Kai Jiau received the B.S. and Ph.D. degrees in electronic engineering from the National Taipei University of Technology, Taipei, Taiwan, in 2012 and 2015, respectively.

He is currently a Principal Research and Development Engineer with the Delta Research Center, Delta Electronics, Inc., Taipei. He was a Post-Doctoral Researcher and a Research Assistant Professor with the National Taipei University of Technology in 2016 and 2017, respectively. His current research interests include artificial intelligence, optimization methods, data science, and applications of mobile application and cloud computing.

Dr. Jiau was a recipient of the Sayling Wen Doctoral Dissertation Award by the Service Science Society of Taiwan in 2015 and the IICM Ph.D. Dissertation Award by the Institute of Information and Computing Machinery in 2015.