2017 IEEE International Parallel and Distributed Processing Symposium Workshops

Multiobjective Vehicle-type Scheduling in Urban Public Transport

David Peña[1], Andrei Tchernykh[2]
Computer Science Department
CICESE Research Center Ensenada
Baja California, México
[1]pdavid@cicese.edu.mx,
[2]chernykh@cicese.mx

Sergio Nesmachnow[3], Renzo Massobrio[4]
Computer Science Institute,
Universidad de la República
Montevideo, Uruguay
[3]sergion@fing.edu.uy
[4]renzom@fing.edu.uy

Alexander Feoktistov[5], Igor Bychkov[6]
Matrosov Institute for System Dynamics and Control Theory of SB RAS, Irkutsk, Russia
[5]agf65@yandex.ru
[6]bychkov@icc.ru

***Abstract*— In this paper, we study the problem of vehicle scheduling in urban public transport systems taking into account the vehicle-type (different capacity and operating cost) known as VTSP. It is modeled as a multiobjective optimization problem (MOP). We propose a heuristic based on MOCell (Multi-Objective Cellular evolutionary algorithm) to solve the problem considering restrictions of government agencies in context of smart cities to improve the Intelligent Transportation Systems (ITS). A set of non-dominated solutions represents different assignments of vehicles to cover trips of a specific route. The conflicting objectives of provider and users (passenger) are to minimize the total operating cost, and maximize the quality of service, reducing the waiting time and congestion in buses. We present experimental analysis and conclude that the proposed heuristic provides a good performance and competitive results in terms of convergence and diversity of the solutions along the Pareto front.**

Keywords- Evolutionary algorithms, Metaheuristics, Multiobjective optimization, public transport, smart cities.

I. INTRODUCTION

Cities around the world are in state of permanent flux and exhibit complex dynamics. A sustainable urban development is a complex problem and has received attention from researches for many decades.

In increasingly interconnected and globalized world, more than half the population (54%) are located in urban areas, unlike 30% in 1950. This abrupt growth implies deep changes in size and distribution of space (people per square meter). This effect will be accentuated in the coming years. An estimated increase in 2050 at 70% of the world population will live in cities [1]. It leads to rise demand for all infrastructures that interact directly with the people, who spread to urban areas hoping to find better job opportunities and a higher quality of life. However, the increase of migrants involves various problems such as congestion, increased demand for a limited supply of natural resources and other types of goods and services including energy, water, education, health and transportation.

The main challenges for cities on urban mobility are often related to the inability of public transport systems to supply needs of a growing number of users. Though each city has different extra issues, authorities and responsible agencies of the mobility share common challenges such as reduce congestion by improving traffic flow, sustainable and cleaner environment, increase the use of public transport, and other greener options such as bikes.

Harrison et al.[2] stated that the term "smart city" denotes an "instrumented, interconnected and intelligent city" [3]. Different areas such as public administration, education, health services, public safety, energy, transportation and logistics can be improved to make more intelligent, interconnected and efficient by computing technologies [4]. Smart cities can reduce costs, make responsible use of resources and encourage the active participation of citizens in decision-making processes, in order to achieve a sustainable and inclusive city.

The Intelligent Transportation System (ITS), also known as smart mobility, are a set of Information and Communications Technologies (ICT) applied for the specific case of transports. Technological advances in computer science and communication systems allow to collect a huge amount of transport and mobility data from citizens and get useful information to make new software tools e.g. interactive systems, algorithms and mobile applications, to benefit users, government organizations and service providers [5].

The main objectives of ITSs are: improve the safety, increase efficiency and capacity, reduce energy consumption and negative environmental impact, enhance economic productivity for users and providers, enhance the personal mobility, convenience, and comfort and create an environment in which the development and deployment of new ITS technologies may appear.

The problems with more than one objective function to be optimized are known as MOPs (Multiobjective Optimization Problems), typically non exist a single solution that optimize all objectives at the same time, a solution of these problems consist of a set of non-dominated solutions, also known as Pareto front or Pareto set. Calculate the Pareto front to solve a MOP in most cases is impractical because it may contain infinite number of non-dominated solutions and there are MOPs that are NP-hard. Therefore, the goal is produce a good approximation of the true Pareto front in reasonable execution time. Heuristics and metaheuristics are needed to find a high-quality solution for MOPs.

This paper presents a heuristic based on multiobjective cellular evolutionary algorithm (MOCell) to solve the vehicle-type and size scheduling problem (VTSP), a variant of the vehicle scheduling problem. The objectives of ITS is conflicting goals due to the provider is seeking to minimize the operating and purchasing cost, but also the users want and expect a better service. Hence, a solution of our algorithm for VTSP proposes a distribution of vehicles (proper frequency calculation) to reduce the operational cost and guarantee the quality of service.

II. Related work

This section presents a brief overview of different models and algorithms for transport problems, particularly urban public transport (Fig.1). Most of these works are based on models proposed decades ago applying computational intelligence techniques to improve approximate solutions, since the problem is NP-hard.

To plan a transport route, it is necessary to solve all the associated problems. To provide solutions, Ceder [6] and Wilson propose activities, usually performed in sequences as follows: (1) network route design, (2) setting frequencies, (3) timetable development, (4) vehicle scheduling and (5) crew scheduling or driver scheduling

It is desirable, therefore, that all five activities be planned simultaneously in order to exploit the system capability to the greatest extent and maximize the system productivity and efficiency.

A. Early heuristic methods

So-called early methods were not very advanced, as nowadays, because computers have not enough power to run complex mathematical solvers and use techniques for mathematical models. Many approaches were reduced to construction of an initial schedule by using heuristic process, and then attempt to improve this schedule by making limited changes.

TRACS (Techniques for Running Automatic Crew Scheduling) was developed at the University of Leeds in 1967 [7]. The system is based on the assumption that a poor initial solution cannot turn into a good solution by heuristic improvements, which might be true since metaheuristics were not available at that time.

B. Mathematical programming methods

IMPACS (Integer Mathematical Programming for Automatic Crew Scheduling) was developed for bus operation in the late 1970s. Parker and Smith presented the prototype and Wren and Smith [8] gave a full description of the system. It was installed in London Transport in 1984 and in Greater Manchester Buses in 1985.

The vehicle or driver scheduling problem can be formulated as a set covering problem and expressed as an Integer Linear Programming (ILP) problem. The basic model is given below: number of potential changes (n), number of work pieces to be covered (m), cost of changes (c_j), $a_{ij} = \{0,1\}$ 1 indicates that the change j covers work piece i, and 0 otherwise. $x_j = \{0,1\}$, 1 indicates that the change j is used in the solution, and 0 otherwise.

The objective is to

$$minimize \sum_{j=1}^{n} c_j x_j$$

subject to:

$$\sum_{j=1}^{n} a_{ij} x_j \ \geq 1, i \in \{1,2,\dots,m\}$$

$$x_j = 0 \ or \ 1 \ , j \in \{1,2,\dots,n\}$$

C. Metaheuristic methods

The techniques for solving combinatorial problems can be classified into two main categories: exact and heuristic algorithms. The exact algorithms guarantee to find the global optimum. However, often only small-sized instances can be practically solved. Heuristics and metaheuristics are more efficient and flexible and allow approximate global optimum.

Baaj and Mahmassani [8] propose a heuristic solution based on the combination of routes, where the initial population is generated from identifying the shortest paths between nodes of high demand. The model includes several restrictions with important issues in ITS.

The objective is:

$$\text{minimize} \left[C_1 \sum_{i=1}^{n} \sum_{j=1}^{n} d_{ij} t_{ij} + C_2 \sum_{j=1}^{n} f_k t_k \right]$$

Subject to:

$$f\_k \geq f\_min \ \ \forall \ k \in R$$

$$LF_k = \frac{Q_k^{max}}{f_k \alpha} \leq LF_{max} \ \forall \ k \in R,$$

$$\sum_{k \in R} N_k = \sum_{k \in R} f_k t_k \leq FS$$

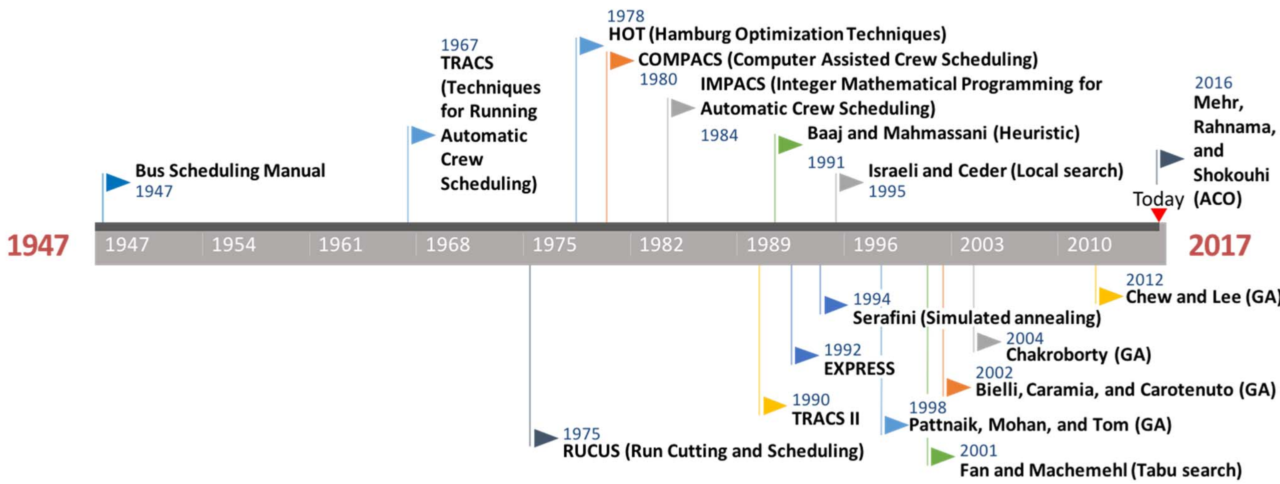


Figure 1. Timeline of evolution of operations research, about study of optimization problems associated with transportation systems.

where

d_{ij}	:	demand between nodes i and j.
t_{ij}	:	total travel time between i and j.
f_k	:	frequency of buses operating on mute k.
f_{min}	:	minimum frequency of buses operating.
t_k	:	around trip time of route k.
Q_k^{max}	:	maximum flow occurring on any link of route.
α	:	seating capacity of buses operating.
LF_k	:	load factor of route k.
LF_{max}	:	load factor of route maximum.
N_k	:	number of buses operating on mute k.
FS	:	fleet size.
R	:	set of routes.
C_1, C_2	:	weights reflecting importance of the two cost.

A local search algorithm proposed by Israeli and Ceder [6], for a multiobjective optimization to minimize the size of the fleet and cost that represents the number of passengers per hour, waiting time of passengers between each stop and travel time when the bus is empty.

$$Min\ f_1 = FS$$

$$Min\ f_2 = \left[C_1 \sum_{i,j \in N} PH_{ij} + C_2 \sum_{i,j \in N} WH_{ij} + C_3 \sum_{r \in R} EH_r \right]$$

Where EH_r is Empty space Hours on route r, FS is Fleet Size and R is a set of routes, PH_{ij} is Passenger Hours and WH_{ij} is Waiting Time between nodes i and j.

The problem has been resolved in three stages. First, several sets of non-dominated solutions are generated, after that the frequencies for each vehicle are determined. Then, local search method is used for exploring solutions. Finally, they are evaluated and the best solutions are selected from the Pareto-optimal set.

A number of GAs have been developed for the vehicle scheduling problem. Pattnaik et al. [9] focus on minimizing the cost associated for both the provider and users. Chakroborty [10], [11] highlights the high effectiveness of genetic algorithms to solve urban transit network design problem.

Shen and Kwan [12] develop an approach called HACS based on a Tabu search for the driver scheduling problem. The HACS is based on a representation of the problem involving sequences of links. The links and its associated active relief opportunities compose a solution space.

Costa et al. [13] present an algorithm in the field of high-speed trains. After random initialization, it uses classical techniques to improve cooling solution and escape from local optimal. The difference with classical simulated annealing algorithms of single objective is the use of weighted aggregation rules of values of objectives.

Recently, Mehr et al. [14] implement a metaheuristic based on ant colony systems for solving the problem of design lines of light rail and bus rapid transit in the city of Mashhad in Iran.

III. The Multiobjective VTSP

Smart city issues propose the development and implementation of computational techniques for planning mobility. ITSs include three main participants:

- Citizens or public transport users looking for an efficient, economical, safe, comfortable and friendly multimodal system with the environment.
- Companies providing transport service, which mostly seek to reduce operating costs and maximize profits, focusing efforts on economic subjects under the regulations of government authorities.
- Governments whose policies seek to ensure a quality of life for its citizens, setting patterns of demand for mobility that meets their needs and ensuring the proper functioning of mobility systems.

We define our problem taking as reference the bus transport systems, which can be extrapolated to other schemes without significant changes.

A. Problem description

The VTSP models a realistic scenario, where a set of vehicles of different types are assigned to the trips to cover a defined route. The optimization problem is to find an appropriate distribution of vehicles, with the goal of simultaneously minimize two important objectives: the operational cost for providers f_1, and unsatisfied user demand f_2 (section III.*B*).

The operational cost (f_1), additionally contributes to minimizing the impact to the environment and improve the traffic flow, due to small vehicles take less space in the road and reduce the fuel consumption.

The unsatisfied user demand effects on the perceived delay to board the vehicle (waiting time) and the comfort associate to load factor, i.e. number of passengers on board.

We follow assumptions proposed by Ceder [6]:

- Assume that the problem network route design is solved, therefore, the route with their stops are defined.
- Must have a passenger demand (load profile) for each timeslot in every stop (e.g. number of passengers in the LA bus route showed in the Figure 2).
- Unsatisfied demand defines the amount of passenger that cannot be moved satisfactory, which implies more waiting time and congestion in the selected vehicles to cover the route in this period.
- The cost for each vehicle-trip include the cost of driver, fuel consumption and vehicle maintenance.

B. Mathematical formulation

Our problem formulation is presented below. Given the following elements: A set of vehicle $B = \{b_1, \dots, b_n\}$, where $b_i$shows the number of vehicles of type i, where n is the number of different types of vehicles and $\sum_{i=1}^{n} b_i$ is the total fleet. T is a set of required trips $T = \{t_1, \dots, t_m\}$ of a defined route R.

The VTSP is based on two objective functions f_1 and f_2 :
Minimize

$$f_1 = \sum_{\mathrm{i=1}}^{\mathrm{n}} \omega_{\mathrm{i}}$$

and

$$f_2 = \sum_{\mathrm{s \in R}} \mathrm{LQ_s}$$

subject to:

$$\mathrm{c_i} = \mathrm{c_i^{bus}} + \mathrm{c_i^{gas}} + \mathrm{c_i^{driver}},$$

$$\omega_{\mathrm{i}} = \mathrm{c_i m_i},$$

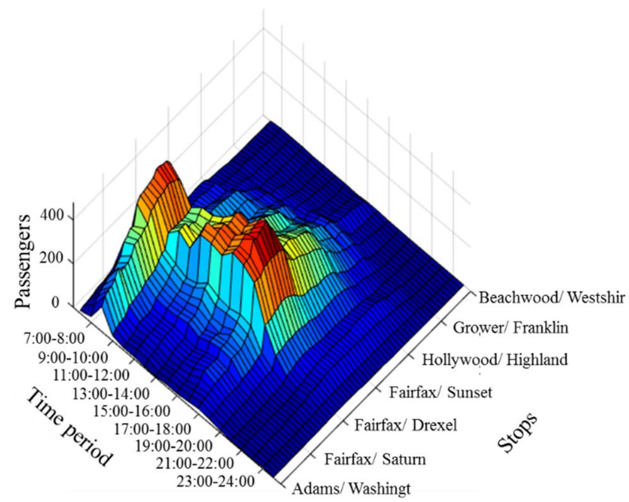
$$\mathrm{f_j} \geq \mathrm{f_{min}},$$

$$\mathrm{LF_j} = \frac{P_j^{max}}{\mathrm{CAP_i} \times \mathrm{f_j}} \leq \mathrm{LF_{max}},$$

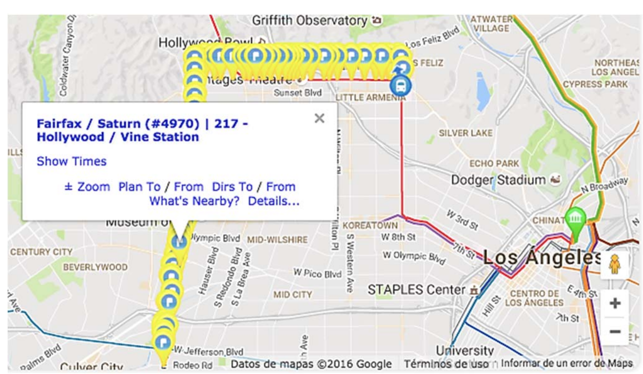
$$\mathrm{LQ_s} = \max\left(P_j^s - \textstyle\sum_{\mathrm{i \in M_j}} \mathrm{LF_j} \times \mathrm{CAP_i}, 0\right)$$

where

$\mathrm{c_i^{gas}}$	: cost of fuel for each vehicle.
$\mathrm{c_i^{driver}}$	: cost hourly pay of the driver.
$\mathrm{c_i^{bus}}$	: cost of maintenance and operation of vehicle.
$\mathrm{c_i}$	: total cost of use of vehicle-type i.
$\mathrm{m_i}$	: number of vehicles-type i to cover trips on T.
ω_{i}	: cost of use vehicles-type i to cover trips of T.
P_j^{max}	: maximum number of passengers at any stop.
P_j^s	: number of passengers on stop s in the route R.
$\mathrm{f_j}$	: frequency for the period j.
$\mathrm{f_{min}}$	: minimum operating frequency.
$\mathrm{LF_j}$	: load factor for the period j.
$\mathrm{LF_{max}}$	: maximum load factor for the period j.
ℓ_s	: distance between the stop s and s-1.
$\mathrm{M_j}$	: set of vehicles used during the period j
$\mathrm{CAP_i}$	: capacity of a vehicle from type i.
$\mathrm{LQ_s}$	: demand for passengers at the stop s that exceed the vehicles capacity.



(a)



(b)

Figure 2. (a) Ride-check data for LA bus route 217 northbound, 19 time-period of one hour and 59 stops, maximum load of passengers 481 in Fairfax/ Rosewood during time-period between 17:00 to 18:00 (peak hour) [6]. (b) Route map with stops placed (southbound and northbound) of route 217 in Los Angeles – California, show times of real-time info for schedules in weekday (Jun 27, 2016 - Dec 9, 2016) [15].

IV. MULTIOBJECTIVE EVOLUTIONARY ALGORITHMS

The problem tackled in this paper is composed of two contradictory objectives that have to be optimized at the same time. The formulation of a multiobjective optimization problem (MOP) [16] is the following.

Find a vector $x^* = [x^*_1, x^*_2, \ldots, x^*_n]$ which satisfies the m inequality constraints $g_i(x) \geq 0, i = 1, 2, \ldots, m$, the p equality constraints $h_i(x) = 0, i = 1, 2, \ldots, p$, and minimizes the vector function:

$$f(x) = [f_1(x), f_2(x), \ldots, f_k(x)]^T$$

Where $x = [x_1, x_2, \ldots, x_n]^T$ is the vector of decision variables.

MOP consists of k objectives reflected in the k objective functions, $m + p$ constraints on the objective functions and n decision variables. The set of all the values satisfying the constraints defines the feasible region (or solution space) S, and any point $x \in S$ is a feasible solution.

A key concept in multiobjective optimization is *Pareto dominance*, which is defined as: given two vectors $u = (u_1, \ldots, u_k)$ and $v = (v_1, \ldots, v_k)$, we say that u dominates v (denote by $u \prec v$) if and only if u is partially less than v, i.e., $\forall i \in \{1, \ldots, k\}, u_i \leq v_i$ and $\exists i \in \{1, \ldots, k\}: u_i < v_i$.

Solving a MOP can be viewed as the process of finding the set of solutions that dominate every other point in the solution space. This means that the solutions in this set are *Pareto optimal* for the problem, a set composed of all the Pareto optimal solutions is known as the *Pareto Optimal Set*, or simply the *Pareto set*. Each vector in the Pareto set has a correspondence in objective function space, leading to the so-called *Pareto front* [17]. Formally:

Pareto Optimality: A solution $x^* \in S$ is Pareto optimal if and only if it is non-dominate by any other solution $x' \in S$

Pareto Optimal Set: For a given MOP, $f(x)$, the Pareto Optimal Set, $\mathcal{P}_S$, is defined as:

$$\mathcal{P}_S = \{x \in S | \nexists x' \in S \; f(x') \prec f(x)\}$$

Pareto Front: For a given MOP, $f(x)$, and Pareto Optimal Set, $\mathcal{P}_S$, the Pareto Front $\mathcal{P}_F$ is defined as:

$$\mathcal{P}_F = \{f(x) \in \mathbb{R}^k | x \in \mathcal{P}^*\}$$

As discussed before, MOPs can have a Pareto front composed by a huge (possibly infinite) number of solutions, we only aim for an approximation of the Pareto front. When using stochastic techniques, such as metaheuristics (e.g. evolutionary algorithms, simulated annealing or Tabu search), the goal is to obtain a Pareto front approximation (also called approximation set), i.e. a subset of solutions that represents the true Pareto front ($\mathcal{P}_F$).

A. Evolutionary algorithms and MOCell

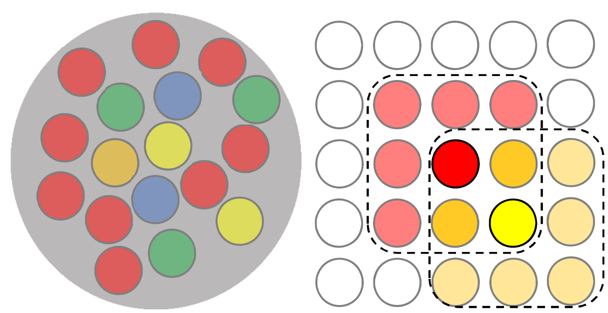
Evolutionary algorithms (EAs) are nature-inspired search methods that mimic the evolution process of species in nature to solve optimization problems. The evolution is the result of the interaction between the creation of new genetic information, its evaluation and future selection. Each individual is affected by other individuals and the environment, with different probabilities depending to the type of configuration of the population, when an individual exhibit a better performance, it has a greater opportunity to live for a longer and generate a genetic inheritance (mutated) by mating. The non-deterministic nature of reproduction leads to a permanent production of new genetic information and therefore to the creation of differing offspring.

The multiobjective EAs (MOEAs) have been applied to solve hard MOPs. They deal simultaneously with a set of possible solutions, obtaining accurate results when solving problems in many research areas (e.g. bioinformatics, transport, structural and mechanical engineering, robotics, scheduling, finance or manufacture process). Due to their population-based nature, they are able to find a set with several solutions, for approximate the whole $\mathcal{P}_F$ of an MOP in one single run.

MOEAs are designed to take into account two features at the same time: satisfactory convergence and diversity properties. This means that they not only seek to find the approximate $\mathcal{P}_F$, with a high degree of convergence (be as close as possible to the $\mathcal{P}_F$), but also the Pareto optimal solutions must be uniformly spread along the $\mathcal{P}_F$.

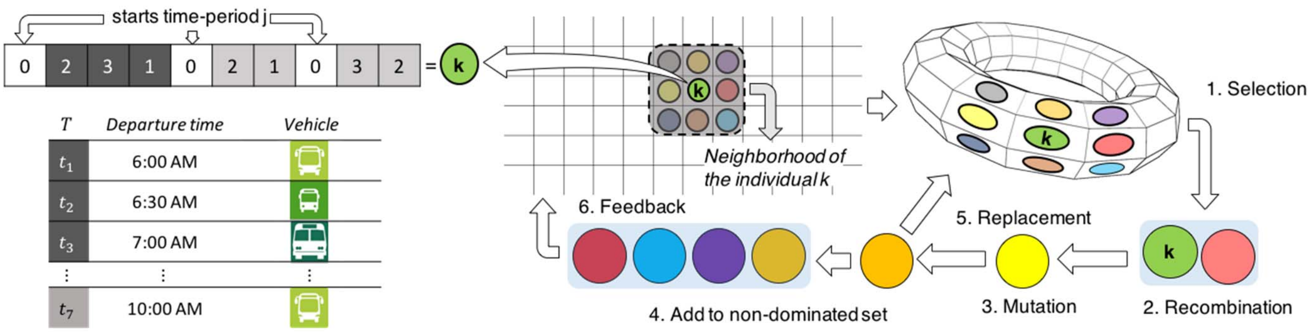
These techniques apply an iterative and stochastic process on a set of individuals (population), where each individual represent a potential solution to the problem. To measure their aptitude in every objective of the problem, the individuals are assigned a fitness value used by the algorithm to guide the search.

Most EAs use a single population (panmixia) of individuals and apply the operators to them as a whole, if we think in the population of an EA in terms of graphs, a panmictic EA is a completely connected graph. On the other hand, in the case of distributed EAs or cellular EAs (cGAs) the individuals in a population can only interact with a reduce number of individuals partitioned in a set of island or located in a nearby neighborhood respectively (Figure 3).



(a) (b)

Figure 3. Example of individual's distribution in a population for (a) Panmictic EA and (b) cGAs.



(a) (b)

Figure 4. (a) Example of solution representation for the VTSP, (b) Reproduction steps in asynchronous MOCell

In this work, we focus on the cGAs, particularly, on MOCell [18]. The main feature of this type of algorithms is that each solution belongs to a cell and can only recombined with neighboring cells distributed in a toroidal grid. The main idea of this limitation is to perform a greater exploration of the search space because the induced slow diffusion of solutions through the population provides a kind of exploration (diversification), while exploitation takes place inside each neighborhood by genetic operator. It maintains an external archive to store non-dominated solutions that is bounded and uses the crowding distance of "Non-dominated Sorting Genetic Algorithm, version II" (NSGA-II) for maintain a diverse set of solutions [16].

B. Encoding and solution representation

Solutions are encoded as arrays of integers, representing the vehicle-type assigned to cover a trip of T. Zeros mark new time-periods. The order of departures is specified in the sequence. Figure 4 shows an example of solution encoding for an instance with 3 different type of vehicles, 7 trips $s_k \in S$, and 3 periods of time of one hour.

The array size is taken from prior demand study and preliminary frequency determination. A method based on a load profile, this considers a lower-bound level on the frequency (F_j) for each time-period, given the same vehicle-capacity constraint, $\overline{\text{CAP}}$ (average of capacity of the different vehicle-type) and it is expressed as follows:

$$F_j = max\left[\frac{A_j}{\text{LF}_\text{j} \cdot \overline{\text{CAP}} \cdot L}, \frac{P_j^{max}}{\overline{\text{CAP}}}, \text{f}_{\min}\right]$$

$$A_j = \sum_{s \in R} \text{P}_{\text{j,s}} \times \ell_\text{s}$$

$$L = \sum_{s \in R} \ell_\text{s}$$

Where A_j is the area in passenger-km under load profile during time period j and L is the route length [6].

The distribution of zeros can be changed but cannot be consecutive and every time-period j has a same length and depends on total travel time to cover the route R.

Algorithm 1. Pseudocode of metaheuristic based on MOCell.

```
[data]=Setup (); /Algorithm parameters and data input
pop=[popGen()&neighborhood()]; /Creates an initial population and distribute in a toroidal grid
𝒫_F= []; /Creates an empty Pareto front
while (terminationCondition==true) do
   for k=1to popSize do /individual = k
      ndPop=getNeighborhood(pop,k);
      parents=selection(ndPop);
      offspring=recombination(data, parents);
      offspring=mutation(data, offspring);
      pop(k)=replacement(ndPop, offspring);
      𝒫_F=insert𝒫_F (pop(k), 𝒫_F);
   end
pop=feedback(paretoFront);
end
```

C. Objective functions and fitness evaluation

The optimization problem is formulated as having two different objective functions f_1 and f_2. The first one is to minimize the operational cost that covers all departures proposed for a specific route, and f_2 is to minimize the perceived loss of quality of service when the user demand is not satisfied.

The f_1 equation (section III.B) consists of the summation of the all cost for each vehicle assigned to cover a trip t_i, which is calculated according to the schedule (timetable) and variable cost depending of the vehicle type. The f_2 indicates the number of passenger-km that cannot be transported by the fleet assigned for each trip t_i.

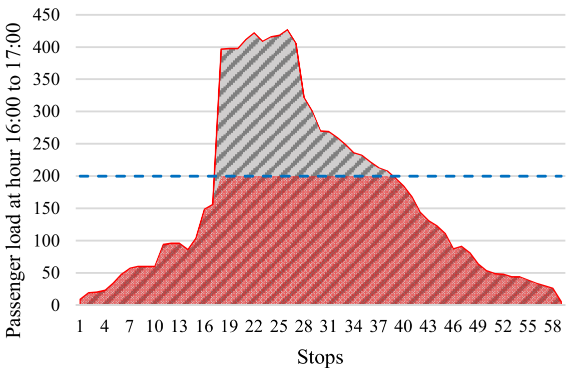


Figure 5. The dotted line is the capacity of the vehicles assigned in a timeslot j. Unsatisfied demand (f_2) defines the number of passengers that cannot be moved satisfactorily (▨), which implies more waiting time and overload in the selected vehicles to cover the route in this timeslot.

The value of the functions is normalized since we know the highest cost of the operation by selecting all the most expensive vehicles. It is also possible to calculate the maximum loss of passengers when vehicles with minor capacity are selected.

D. Evolutionary operators

Population initialization: the population is generated by randomly assigning different types of vehicles to each departure taking into account the size of schedule and zeros distribution previously defined, after that, distributed all individuals in a toroidal grid of 10x10 (Figure 4).

Selection: a tournament selection (tournament size: 8 individuals) to select the parents in a neighborhood of a study individual in the (x, y) position.

Recombination: a classic cut and crossover technique of recombination is used, because it preserves the order in a good solution and encourages elitism.

Mutation: we use swap mutation, selects a randomly set of vehicles and exchanges them with a different type.

V. Experimental Results

In this section, we detail the experimentation methodology we have used in our study. First, we describe the quality indicators used to assess the quality of the computed Pareto front. Second, we report the parameter settings and experimental analysis of the proposed algorithm for an example route 217 from LA City (Figure 2).

A. Quality indicator

Different metrics have been proposed in the literature to evaluate MOEAs. We have chosen the Hypervolume [19] quality indicator, which evaluate convergence and maximum spread at the same time. It calculates the hypervolume (Hv) of the multidimensional region enclosed by the individuals in the computed approximation to the Pareto front and a "reference point" in the objective function space (Figure 8 (c)). The closer the approximation is to the Pareto optimal front, the higher the value of this indicator. On the other hand, if the spread of the individuals along the Pareto front is good (desirably uniform), the higher the value of this indicator. Hence, a solution that produce the higher value as possible of this indicator is desirable.

The hypervolume can be calculated as follows. First, given an approximation set placed in objective space and a reference point, a hypercube (Hc_k) for each solution (k) is constructed, taking as corners the reference point and the solution point. After that the union of all these hypercubes is equal to the hypervolume, mathematically:

$$Hv = \bigcup_{k=1}^{popSize} Hc_k$$

B. Experimentation methodology

In this section, we describe an evaluation of the proposed MOCell algorithm focusing on quality of solutions and performance. For the considered problem instance, we run 30 independent executions of 10000 fitness evaluation. After this, Hv has been calculated for each of the runs.

The experimental analysis was performed on an Intel Core i5 @ 1.6Ghz, 4GB RAM 1.6GHz DDR3 with 64 bit macOS Sierra Version 10.12.1.

C. NSGA-II

To compare the proposed algorithm with other technique for multiobjective combinatorial optimization, we implement a "classical" evolutionary algorithm NSGAII, proposed by Deb in 2002 [20].

The same parameters are used in recombination and mutation operators. The size of the initial population is 100 individuals and 5000 generations on each run, assessing the performance by using the Hv as a quality indicator.

D. Results and discussion

Figure 6 shows an observed cost for each trip of vehicles of the same type [6], and a set of obtained costs. We see that for each time period, obtained operational cost is less for all trips. However, the quality of service worsens as operating costs decrease.

Figure 7 shows timetables obtained by selecting different solutions of the Pareto front approximation for the route 217 in Los Angeles.

TABLE I. PARAMETRIZATION OF THE ALGORITHM

Stopping Condition	10000 function evaluation
Population Size	100 individuals (10x10)
Neighborhood	8 surrounding neighbors
Selection Parents	Binary tournament
Recombination	Crossover (cut only in time-period start)
Probability of recombination	0.5
Mutation	Swap
Probability of mutation	1
Replacement	Replace if better ranking and crowding
Density estimator	Crowding distance
Feedback	20 individuals

The quality of solutions is presented in Figure 8.

Figure 8 (a) shows the result obtained by the MOCell algorithm after 10000 fitness function evaluations (f_1 y f_2) and compare with the initial population for this run. Figure 8 (b) present a comparison for the non-dominate sets in different generations in a single run of the algorithm, improving as the population evolves, showing a competitive behaviour of the recombination and mutation operators to explore the search space and achieve an approximation to the real Pareto front.

The values obtained after applying Hv to each fronts collected from 30 independent runs show a good performance of the proposed algorithm with maximum Hv value of 0.3139 (0.2991 on average in the last generation).

Figure 8 (c) shows how the non-dominated set generates a hypervolume (union of the hypercubes between each solution and the reference point $x = 1$ and $y = 1$) growing in each generational exchange, highlighting the characteristic elitism of MOCell and the effect of feedback from the betters individuals based on the crowding distance [20] which influences in increasing the spreading along the front, hence a greater hypervolume. We observe that each run improves the hypervolume in a similar way as shown in Figure 8 (d), providing a stable behavior for the execution parameters of the algorithm.

Figure 9 shows the $\mathcal{P}_F$ approximation obtained in a better run for NSGA-II ($Hv = 0.2735$) and our algorithm ($Hv = 0.3139$) for the same instance (LA route 217). Furthermore, we compare the non-dominate set formed by the 30 best approximation sets for each run. We analyze the Pareto front approximations for the two algorithms. It is easy to see that NSGA-II has difficulties in convergence and diversity, hence it has poor Hv value.

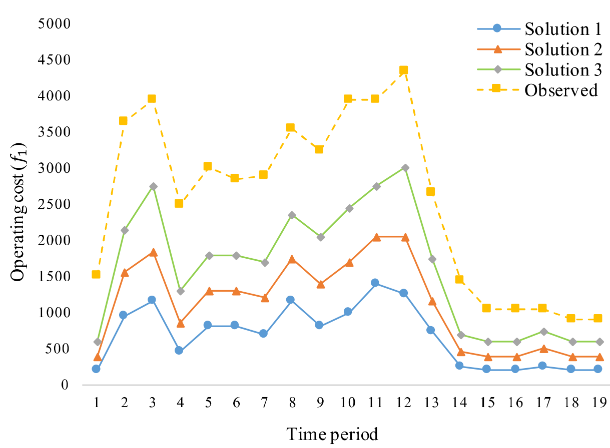


Figure 6. Comparison of operational cost results of three different solutions (Pareto front ends and random individual) and the observed frequency of route 217 in Los Angeles with only one type of vehicle with capacity for 90 passengers.

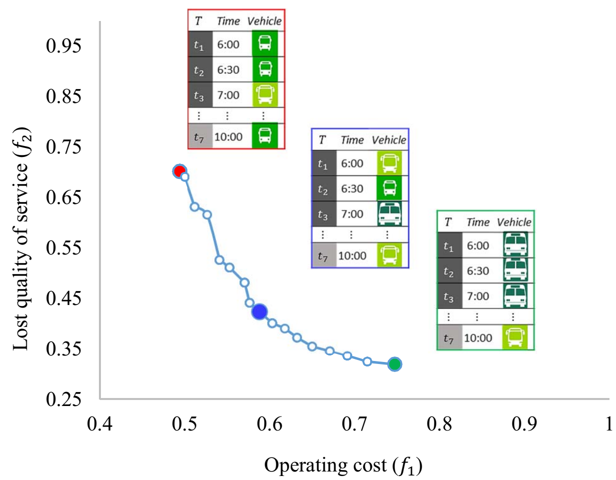


Figure 7. Timetables obtained by selecting different solutions of the Pareto front approximation for the route 217 in Los Angeles.

The comparison shows that the cellular algorithm is able to improve the NSGA-II results, achieving better hyper volumes values in average and in best case. It provides a good approximation to the optimal Pareto front, and solution sets that preserve diversity.

In real-world applications, the decision maker is normally interested in certain types of trade-offs based on regulations and restrictions usually framed by ITS. From this point of view, the approximation sets produced by our algorithm in Figure 8 (a) (b) repeat a lot of values for the fitness functions.

It happens because buses have the identical timetable but the order of the vehicles is not the same. The decision maker can choose one of the two solutions arbitrarily, when through a detailed study (e.g. local search) in specific times of the day (e.g. peak hours), can helping to select the best of both schedules.

VI. Conclusions

In this paper, we have studied the multiobjective VTSP. The problem has been formulated by considering two conflicting objectives: operational cost and quality of service for users. After describing the problem formulation in details and presenting the chosen algorithm, we explain a method to solve VTSP based on MOCell because it is a very competitive technique for MOP.

The experimental analysis demonstrates the capacity of the studied heuristic to find a set of different vehicle types to cover a specific route with a reduced cost in comparison to other methods. It also guarantees the quality of service defined by government entities.

We can conclude that multiobjective metaheuristics are very useful tool for problems associate to vehicle scheduling, as they are able to provide a wide range of trade-off timetables applicable in different cases taking into account characteristics and regulation of the ITSs.

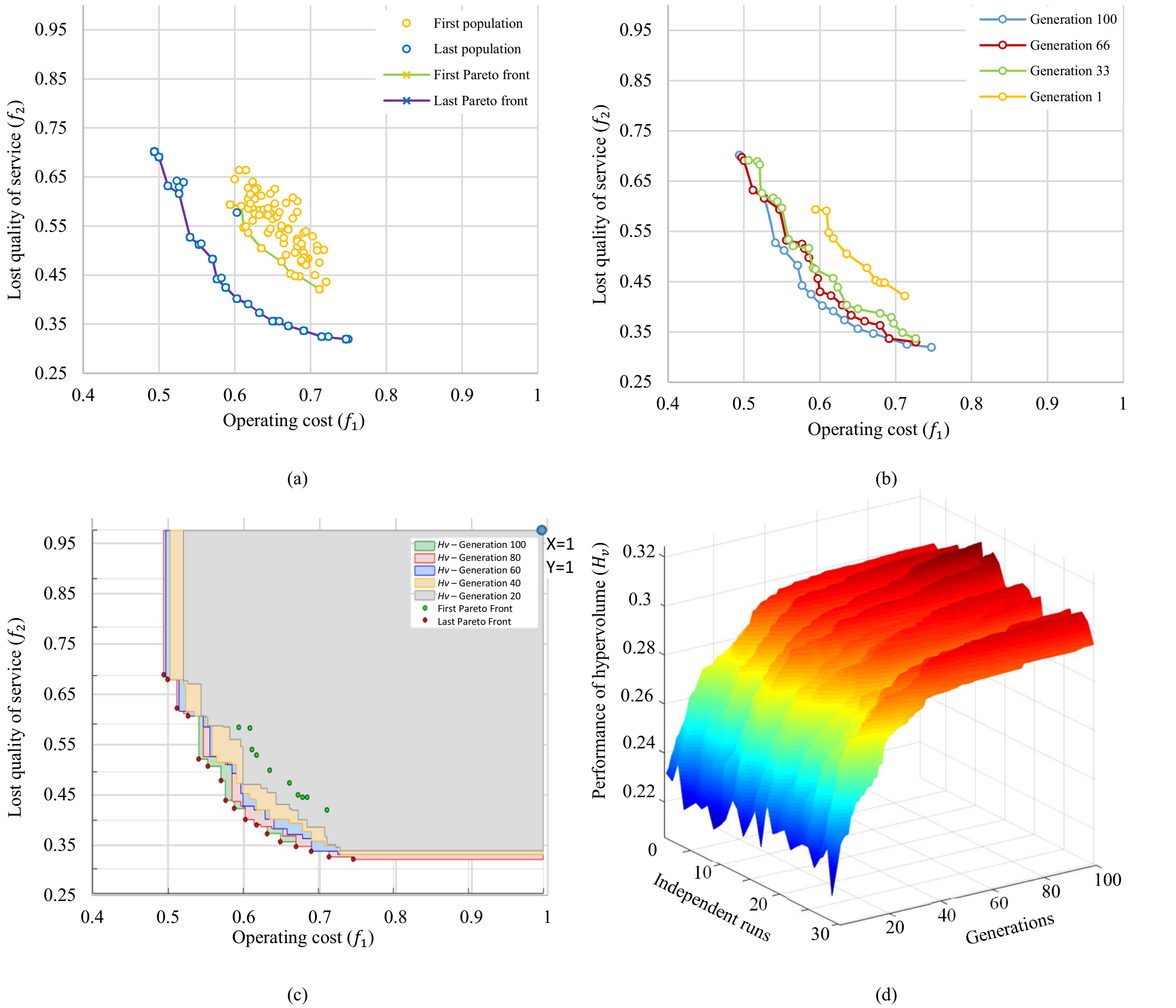


Figure 8. (a) Pareto front approximation corresponding to the $Hv = 0.3139$ in comparison of the initial population results for one problem instance. (b) Performance of Pareto front approximation, for one execution of the proposed algorithm with 100 generations. (c) Hypervolume constructed by the solution and the reference point ($x = 1, y = 1$) across the evolution process in the algorithm. The initial Pareto front is compared with the approximation found in the last generation of the best execution. (d) Performance of hypervolume in 30 independent runs of algorithm and 100 generations i.e. look at the toroidal mesh, individual per individual.

The main lines of future work include a crew scheduling and scheduling that takes into account real-time traffic information to improve the assignation of vehicles and to produce adaptive solutions for each period of time.

Another interesting research is to use the knowledge of the problem to adjust the operators of recombination or mutation, to provide more appropriate scheduling of vehicles of different type, through the implementation of more efficient and effective search methods that allow speed up the optimization process.

ACKNOWLEDGMENT

The work is partially supported by RFBR, project no. 16-07-00931, Council for Grants of the President of Russian Federation, project NSh-8081.2016.9, and CONACYT, México, grant no. 178415.

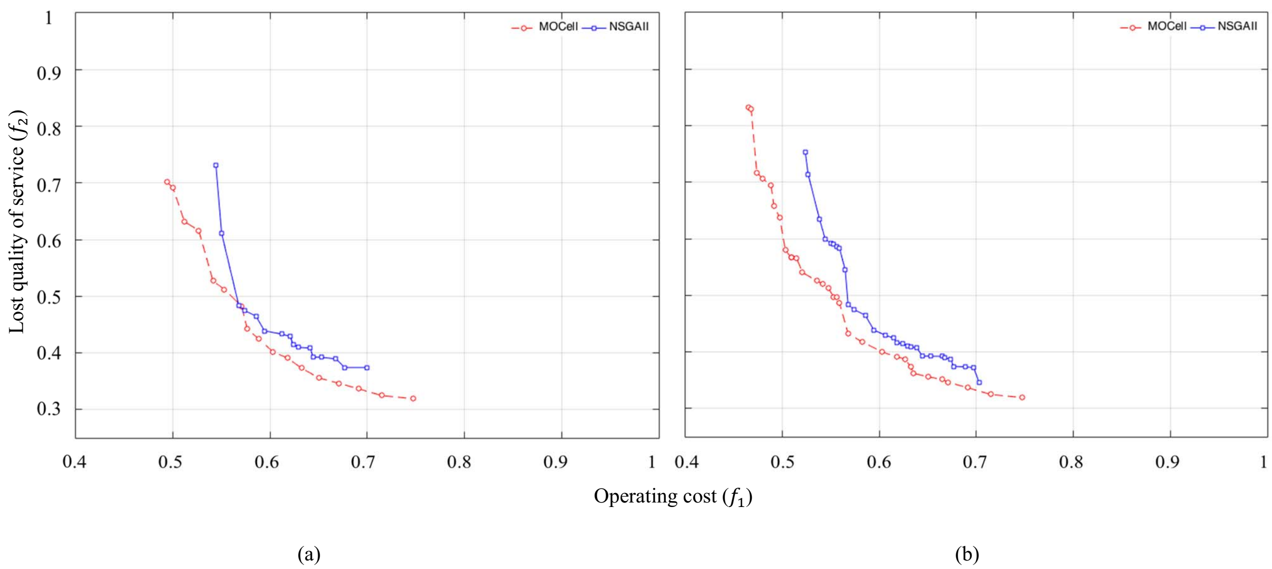


Figure 9. (a) Comparison of best case for NSGA-II and cellular algorithm proposed. (b) Pareto front approximation corresponding to the non-dominate set of 30 best case sets for the route 217 in Los Angeles.

REFERENCES

[1] United Nations, *World Urbanization Prospects: The 2014 Revision, Highlights (ST/ESA/SER.A/352)*. 2014.

[2] C. Harrison, B. Eckman, R. Hamilton, P. Hartswick, J. Kalagnanam, J. Paraszczak, and P. Williams, "Foundations for Smarter Cities," *IBM J. Res. Dev.*, vol. 54, no. 4, pp. 1–16, Jul. 2010.

[3] V. Albino, U. Berardi, and R. M. Dangelico, "Smart Cities: Definitions, Dimensions, Performance, and Initiatives," *J. Urban Technol.*, vol. 22, no. 1, pp. 3–21, Jan. 2015.

[4] D. Washburn and U. Sindhu, "Helping CIOs Understand 'Smart City' Initiatives," *Cambridge, MA Forrester Res. Inc.*, 2010.

[5] R. Gordon, *Intelligent Transportation Systems*. Cham: Springer International Publishing, 2016.

[6] A. Ceder, *Public Transit Planning and Operation*, 1st ed. Burlington: Elsevier, 2007.

[7] J. Li, "Fuzzy Evolutionary Approaches for Bus and Rail Driver Scheduling," The University of Leeds, 2002.

[8] M. H. Baaj and H. S. Mahmassani, "An AI-based approach for transit route system planning and design," *J. Adv. Transp.*, vol. 25, no. 2, pp. 187–209, Mar. 1991.

[9] S. B. Pattnaik, S. Mohan, and V. M. Tom, "Urban Bus Transit Route Network Design Using Genetic Algorithm," *J. Transp. Eng.*, vol. 124, no. 4, pp. 368–375, Jul. 1998.

[10] P. Chakroborty, "Genetic algorithms for optimal urban transit network design," *Comput. Civ. Infrastruct. Eng.*, vol. 18, pp. 184–200, 2003.

[11] P. Chakroborty, "Brics Optimal Routing and Scheduling in Transportation : Using Genetic Algorithm to Solve Difficult Optimization Problems," *Dir. Inst. tech- Nol. kanpur*, p. [S.l.], 2004.

[12] Y. Shen and R. S. K. Kwan, "Tabu Search for Driver Scheduling," in *Computer-Aided Scheduling of Public Transport*, S. Voß and J. R. Daduna, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001, pp. 121–135.

[13] A. L. Costa, M. C. Cunha, P. A. L. F. Coelho, and H. H. Einstein, "Application of the Simulated Annealing Algorithm for Transport Infrastructure Planning," in *Metaheuristics and Optimization in Civil Engineering*, X.-S. Yang, G. Bekda\cs, and M. S. Nigdeli, Eds. Cham: Springer International Publishing, 2016, pp. 235–253.

[14] H. S. Mehr, M. R. Rahnama, and M. A. Shokouhi, "Optimization of Main Public Transport Paths Based on Accessibility — Case Study : Mashhad , Iran," vol. 19, no. 1, pp. 114–128, 2016.

[15] Esidian, "Rideschedules." [Online]. Available: https://rideschedules.com/schedule.html?23467. [Accessed: 01-Nov-2016].

[16] G. Zavala, A. J. Nebro, F. Luna, and C. A. C. Coello, "Structural design using multi-objective metaheuristics. Comparative study and application to a real-world problem," *Struct. Multidiscip. Optim.*, no. 2016, pp. 1–22, 2015.

[17] C. a C. Coello, G. B. Lamont, and D. a Van Veldhuizen, *Evolutionary Algorithms for Solving Multi-Objective Problems*. Boston, MA: Springer US, 2007.

[18] A. J. Nebro, J. J. Durillo, F. Luna, B. Dorronsoro, and E. Alba, "MOCell: A cellular genetic algorithm for multiobjective optimization," *Int. J. Intell. Syst.*, vol. 24, no. 7, pp. 726–746, Jul. 2009.

[19] E. Zitzler and L. Thiele, "Multiobjective Evolutionary Algorithms : A Comparative Case Study and the Strength Pareto Approach," vol. 3, no. 4, pp. 257–271, 1999.

[20] 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, 2002.