Comparison of Multiobjective Evolutionary Algorithms for Solving The Multiobjective Route Planning in Dynamic Multi-hop Ridesharing

Wesam Herbawi and Michael Weber
Institute of Media Informatics
University of Ulm
Ulm, Germany
Email:{wesam.herbawi, michael.weber}@uni-ulm.de

Abstract—**Ridesharing is considered as one of the promising solutions for dropping the consumption of fuel and reducing the congestion in urban cities, hence reducing the environmental pollution. Route planning is a key component for the success of ridesharing systems in which multiple objectives can be optimized. The multiobjective route planning problem in multi-hop ridesharing is categorized as NP–complete. Multiobjective evolutionary algorithms have received a growing interest in solving the multiobjective optimization problems. In this work, we compare the behaviour of different multiobjective evolutionary algorithms for solving the multiobjective route planning in dynamic multi-hop ridesharing.**

Comparison results indicate that there is no single algorithm, as in literature, that wins all the tournaments regarding all the quality indicators. However, a subset of the algorithms is recommended with better quality and runtime.

I. Introduction

In recent decades the global warming has taken the global temperature to its highest level in the past millennium and there is a growing consensus that increasing anthropogenic greenhouse gases participates in the global warming [1]. According to [2], in 2006, 51 percent of the liquids made from petroleum and biomass is used by transportation and this share is expected to rise up to 56 percent in 2030. In addition, as in 2006, transportation accounts for 29 percent of total U.S. greenhouse gases emissions [3].

According to a study performed by the European Environment Agency taking into consideration 12 European countries and covering the period from 2004 to 2008 [4], the average car occupancy ranges from 1.18 to 2 passengers per car. USA has no better occupancy rates [5] and this leads to increase the congestion. It is estimated that the cost of lost hours and wasted fuel, resulted from congestion in the USA was 78 billion dollars in 2007 [6].

In their work Jacobson and King [7] have showed that adding 1 passenger to every 10 traveling cars would potentially save 7.54-7.74 billion gallons of fuel per year. A straight forward solution could be the ridesharing which is the shared use of a car by its driver and one or more passengers called riders. These days, there are many ridesharing services and the demand for these services has increased sharply in recent years.

The growing ubiquity of the internet enabled mobile devices enables practical dynamic ridesharing [8]. By dynamic ridesharing, we refer to a system where the process of matching the riders and drivers to form ridesharing is done on very short notice or even en-route [9]. In the USA, the number of persons who use their mobile devices to access the internet has doubled from 2008 to 2009 to reach 63 million people in January 2009, a third on daily basis [1].

A dynamic ridesharing system in which the rider is matched with one or more drivers is called Dynamic Multi-hop Ridesharing System (DMR). DMR is more flexible than other forms of ridesharing and offers more choices for the rider [10]. For the rider, the set of drivers' offers that constitute a trip are called route plan. Route plans in DMR are subject to multiobjectives such as the minimization of time and cost which leads to multiobjective route planning. Usually a single route plan does not minimize all objectives in parallel especially in the presence of conflicting objectives [11].

Multiobjective route planning is categorized as NP-complete [12]. The Generalized Label Correcting algorithm (GLC) [13] is a deterministic algorithm that can be used to find the set of optimal (known as Pareto-optimal) set of route plans, but its exponential worst case complexity is prohibitive.

Multiobjective evolutionary algorithms (MOEAs) have received a growing interest in solving the multiobjective optimization problems especially for their scalability as compared with the deterministic algorithms [14]. Many multiobjective evolutionary algorithms have been proposed in the literature but there is still no consensus about the best algorithm. Actually the behavior of the algorithms is problem dependent and in many algorithms' comparisons on benchmark problems there is no one algorithm that wins for all the problems [15]–[18].

In this work, we compare the behavior of different state of the art MOEAs in solving the multiobjective route planning problem in DMR. The objectives to be minimized are the cost, time and the number of drivers in the route.

[1] comScore: Mobile Internet Becoming A Daily Activity http://www.comscore.com, accessed on 12 August 2010.

978-1-4244-7835-4/11/$26.00 ©2011 IEEE

We utilize the jMetal framework [19] in the comparison. jMetal provides a rich set of classes of multi-objective metaheuristics in addition to some experimentation tools and quality indicators.

The rest of this paper is organized as follows. We introduce some basic concepts in Section II. In Section III we describe the modeling of drivers' offers for route planning and in Section IV we overview the different algorithms to be compared. The problem specific evolutionary algorithmic details and the objective functions are described in Section V. Finally we describe the experimentation method and discuss the results in Section VI.

II. Preliminaries

In this section, we define some concepts used throughout this work.

We suppose that the database has a set of drivers $D = \{d_1, d_2, ..., d_z\}$, a set of stations $S = \{s_1, s_2, ..., s_m\}$ and a set of drivers' offers $O = \{o_1, o_2, ..., o_k\}$. Each offer consists of a starting station s_1, ending station s_2, departure time t_1, arrival time t_2, driver d and cost c such that $\forall o \in O : o = \{s_1, s_2, t_1, t_2, d, c : s_1, s_2 \in S \wedge d \in D \wedge t_1 < t_2 \wedge s_1 \neq s_2\}$.

A request q for sharing a ride contains a source station s_1, destination station s_2 and departure time t such that $\forall q : q = \{s_1, s_2, t : s_1, s_2 \in S \wedge s_1 \neq s_2\}$

Definition 1 (Route Plan in DMR): Is the set $p \in P$ of drivers' offers that are sufficient to serve a request q such that:

$P = \{\{o_1, o_2, ..., o_n\} \subseteq O : s_2(o_i) = s_1(o_{i+1}) \wedge t_2(o_i) \leq t_1(o_{i+1}) \wedge s_1(o_1) = s_1(q) \wedge s_2(o_n) = s_2(q) \wedge t_1(o_1) \geq t(q) \forall o_{i \in \{1,...,n-1\}}\}$

The cost function, also called objective function, $C : P \rightarrow [\mathbb{R}^+]^\omega$ assigns the ω dimensional cost vector $(c_{i,1}, c_{i,2}, ..., c_{i,\omega})$ for route plan $p_i \in P$.

Definition 2 (Pareto-dominance): For two route plans $p_1, p_2 \in P$. We say that p_1 dominates p_2, written $p_1 \prec p_2$, iff $\forall c_{1,j} \wedge c_{2,j} : c_{1,j} \leq c_{2,j} \wedge \exists c_{1,j} < c_{2,j} \; j \in \{1, 2, ..., \omega\}$. In other words $p_1 \prec p_2$ if p_1 has at least one objective value less than the corresponding value of the same objective in p_2 and the other objectives values are not more than their corresponding values in p_2.

Definition 3 (Pareto-optimal): If $\forall x \in P : \nexists x \prec p$ the route plan p is called Pareto-optimal.

Definition 4 (Pareto-optimal Set): Is the set of all Pareto-optimal route plans. Formally, Pareto-optimal set of route plans=$\{p : \forall x \in P : \nexists x \prec p\}$.

Definition 5 (Multiobjective Route Planning in DMR): Is the process of finding the Pareto-optimal set of route plans with respect to request q.

This multiobjective route planning in DMR differs from the classical scheduling problems as following. If we consider the travel from one station to another to be an operation, then in this problem the set of operations is not known in advance as in classical scheduling problems and has to be planned. In addition, in this problem the operation time is not fixed and dependes on the resource to be used if we consider the offers to be resources in analogy to the scheduling problems.

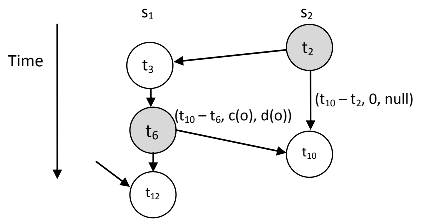

Fig. 1. Simple time-expanded graph. White vertices represent arrival events and gray vertices represent departure events

III. Modeling Drivers' Offers for Route Planning

In this section we discuss how we model the drivers' offers to solve the route planning problem.

The drivers' offers are represented as a simple time-expanded graph [20] G= (V, E). The vertices in the graph refer to the departure and arrival events. For each offer o, two vertices $u, v \in V$ are introduced, the first represents the departure event from station s_1 at time t_1 and the second represents the arrival event to station s_2 at time t_2. Each vertex is annotated with its associated station s and event time (timestamp) t.

An edge $e = (u, v) \in E$, is inserted between each two departure and arrival vertices to represent the travel from s_1 to s_2 with weight vector $w(e) = (time : t, cost : c, driver : d) = (\Delta(t_1(o), t_2(o)), c(o), d(o))$. Note that we have considered the driver as part of the edge weight as it will be used to compute the number of drivers for the route plan.

For each station s, the set of vertices that represent either departure or arrival events are sorted in ascending order regarding their timestamps. An additional edge between each two consecutive vertices $e = (v_i, v_{i+1})$ that belong to the same station s is inserted to represent the waiting at a particular station s with $w(e) = \{\Delta(t(v_i), t(v_{i+1})), 0, null\}$.

Figure 1 shows an example of simple time-expanded graph model of drivers' offers where s_1 and s_2 are two stations. At each station the gray vertices represent departure events to their corresponding arrival events, white vertices, at the other station. The gray vertex at station s_1 represents a departure event from s_1 at time t_6 to station s_2 at time t_{10} with the cost vector annotating the edge connects them.

Given that the drivers' offers are modeled as a time-expanded graph, the multiobjective route planning problem is reduced to solving the known multiobjective shortest path problem (MSPP) on a time-expanded graph. A route plan will be equivalent to a path from source vertex to destination vertex.

IV. Algorithms Overview

In this section, we provide the algorithmic details of the different evolutionary algorithms to be compared. All of the algorithms are genetic algorithms except PAES which is an evolution strategy algorithm. We consider state of the art MOEAs

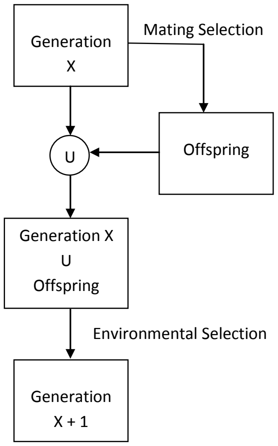

Fig. 2. General Skeleton of The Genetic Multiobjective Evolutionary Algorithms. ∪ is the union all operator

for the comparison which are: Non-dominated Sorting Genetic Algorithm (NSGA-II) [17], Improved Strength Pareto Evolutionary Algorithm (SPEA2) [21], Region-based Selection in evolutionary multiobjective optimization (PESA-II) [22], Indicator-Based Selection in Multiobjective Search (IBEA) [23], A Dynamic Population Sizing Approach for Solving Expensive Multiobjective Optimization Problems (FastPGA) [16], Pareto Archived Evolution Strategy (PAES) [18] and a Cellular Genetic Algorithm for Multiobjective Optimization (MOCell) [15].

Following with the help of Figure 2 we describe the general skeleton of the genetic multiobjective evolutionary algorithms based on which we show the major differences between the different algorithms.

Given a set of solutions called population that represents a specific generation (generation x in Figure 2), each solution is assigned a fitness value to indicate how good is that solution. Individuals from this population are selected to participate in the creation of the offspring using mating selection. To make sure that the better solutions are not lost through generations, i.e. to maintain elitism, the population and its offspring are mixed and among them an environmental selection process is made to determine the individuals to pass to the next generation (generation x+1 in Figure 2). Mostly mating selection is randomized where two parent solutions are randomly selected for the creation of child solutions. Environmental selection is mostly deterministic and depends on the solution's fitness.

The first five algorithms namely NSGA-II, SPEA2, PESA-II, IBEA and FastPGA share this general skeleton with major differences in the definition and calculation of the solution's fitness that plays a major role especially in the environmental selection. Regarding the remaining two evolutionary algorithms PAES and MOCell, they are slightly different in the sense that MOCell is a cellular genetic algorithm that divides the population into regions to be treated equally independent of the other regions. PAES is not a genetic algorithm, but an evolution strategy that combines the concept of local search with Pareto-optimal archived population used to judge the selection process.

NSGA-II uses ranking and crowding distance for fitness definition and calculation. It divides the mixture, generation(x) and offspring, into ranks such that the set of solutions that are not dominated by any solution are in rank zero, and those that are dominated by only one solution in rank 1 and so on. Within each rank, individuals are differentiated by the crowding distance which indicates the extent that the objective space area in which the solution resides is crowded by other solutions.

Binary tournament is utilized for mating selection in NSGA-II which among two randomly selected solutions prefers the one that dominates the other. If no one dominates the other, it selects the one in the less crowded area and randomly if their crowding distance is the same. For environmental selection, NSGA-II selects the solutions with the higher rank as first priority followed by the crowding distance to select from the same rank.

SPEA2 assigns fitness for the mixture as follows: the fitness of an individual is the sum of the strengths of the individuals dominating it plus its density. The strength of an individual is the number of solutions it dominates and the density is a decreasing function to the k-th nearest neighbor distance. SPEA2 utilizes binary tournament for mating selection that compares the two selected solutions using only the dominance comparison and randomly selects one if no solution dominates the other. Solutions are sorted in the mixture according to their fitness and the environmental selection selects the fittest solutions to proceed to the next generation.

PESA-II is slightly different in that the mating selection is region based instead of individual based and it keeps an external archive to maintain the set of nondominated solutions found so far. The archive is divided into hyperboxes or regions to which the fitness is assigned based on the number of solutions in each hyperbox. For mating selection, it randomly selects two occupied hyperboxes and randomly selects an individual from the less occupied one. The solutions of the offspring are added to the archive such that for environmental selection, simply keep the nondominated solutions in the archive. If the archive is full and a solution is to be inserted, then if it belongs to the most populated hyperbox, it is rejected. Otherwise, a solution from the most populated hyperbox is removed.

IBEA defines the fitness in terms of binary quality indicator. There are more than one binary quality indicators that can be used. In this study, we use a binary quality indicator based on the hypervolume concept. Each solution in the mixture is assigned a fitness value that indicates the loss in the hypervolume covered by the mixture if the solution is removed. For mating selection, two solutions are randomly selected and the one with the better fitness is selected. For environmental selection, remove the worst solutions from the mixture to match the required size.

FastPGA uses an approach similar to NSGA-II for fitness assignment. It divides the mixture in two ranks; the first rank is the set of all nondominated solutions and the second one contains all dominated solutions. For the first rank, crowding distance is used as fitness to distinguish between the nondominated solutions. The fitness of the individual in the second rank is the summation of the strengths (same as SPEA2 strength definition) of the solutions it dominates minus the strengths of the solutions dominating it. For mating selection, binary tournament is utilized where among two randomly selected solutions the one with the better rank is selected and if they have the same rank then the one with the lower fitness or randomly otherwise. For environmental selection, the mixture is sorted according to the rank and within the rank according to the fitness and the better solutions are selected for the next generation. Another distinguishing point for FastPGA is the variable population size such that the population size is not fixed throughout the generations and depends on the number of the solutions in the first rank.

MOCell is a cellular genetic algorithm with a major difference from the previous algorithms in that the population is arranged in a cellular structure. For each solution, current solution, in the population, two parents from its neighborhood are selected for mating using binary tournament where the neighbors are ranked and assigned crowding distance like NSGA-II. The selected parents are crossed and the resulting child is mutated. If the resulting child dominates the current solution then it replaces the solution in the population and is inserted in an archive for environmental selection. The archive contains only nondominated solutions and if it is full then crowding distance is used to delete a solution from the most crowded area. If the child does not dominate the current solution and is not dominated by it then the child is inserted in to the archive and the current solution's neighborhood is updated to consider the new child if it is better than one of the neighbors. Besides the population, archive members sometimes and randomly participate in the mating selection.

PAES is an evolution strategy that utilizes the idea of the local search with the archiving process. Initially a random solution is generated and inserted into the archive. A copy of the solution, original solution, is then mutated and the resulting new solution is compared with the original one. If it dominates the original solution then it is inserted into the archive and used for the generation of new solution through mutation. If no one dominates the other, then the new solution is inserted into the archive and a comparison between the new solution and the original one is made using the archive to determine the solution that lies in the less crowded area to be selected for generating new solution.

V. Problem Specific Algorithmic Details

In this section, we provide the problem dependent algorithmic details. We present the solution representation, genetic operators and the objective functions.

A. Solution Representation and Population Initialization

A solution is represented as a list L of vertices forming a path from the source vertex to the destination vertex. Given that different paths have different lengths, we followed the variable-length solution representation approach.

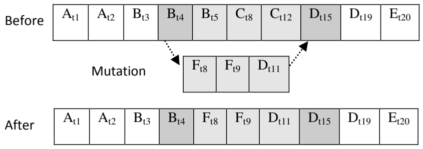

Fig. 3. Path Mutation: dark gray vertices represent the local source and destination vertices and the light gray vertices represent a different path connecting them

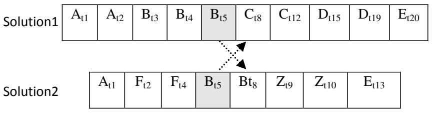

Fig. 4. crossover: gray vertices represent crossover points

Given a request q for a trip from source station $s_1(q)$ at time $t(q)$ to a destination station $s_2(q)$, the list of vertices at station $s_1(q)$ is ascendingly searched for a vertex representing a departure event with a timestamp t being greater than or equal $t(q)$. From that vertex, we utilize the Random Walks (RW) [24] to find a path on the time–expanded graph to a vertex at the destination station $s_2(q)$. This process is repeated to initialize the population.

We utilize the RW for population initialization because this method is characterized by generating more diverse paths as compared with other approaches such as A*, depth-first, and breadth-first and enables the MOEA to well explore the search space [25].

During population initialization and offspring creation, a station might be visited twice. We suppose that a path with a duplicate station will have higher values for the objective functions and we simply consider it as a penalty.

B. Genetic Operators

In this section, we describe our crossover and mutation operators customized for the problem at hand.

We make mutation by means of local search to create an alternative local path in the selected solution. We randomly select two vertices from L and then RW is utilized to find an alternative path connecting them. Figure 3 clarifies the proposed mutation operator.

For crossover, we make a single point crossover between each two selected solutions. A vertex from one solution is randomly selected and a search for a matching vertex in the second solution is made. If a matching vertex is found, then the solutions are crossed after the matching vertices as shown in Figure 4. We noticed that the match rate is low because of the time-expansion nature of the graph, so if a match is not found, we keep searching for another match until all match possibilities are explored.

TABLE I
ALGORITHMS SETTINGS

	NSGA-II	SPEA2	PESA-II	IBEA	FastPGA	MOCell	PAES
Archive Size	-	100	100	100	-	100	100
Max Population Size	-	-	-	-	100	-	-
crossover Probability	1.0	1.0	1.0	1.0	1.0	1.0	-
Mutation Probability	0.4	0.4	0.4	0.4	0.4	0.4	1.0
a, b, c & d	-	-	-	-	20, 1, 20 & 0	-	-

C. Objective Functions

Equations 1-3 explain the calculation of the objective functions where the solution is represented as a list L of vertices. For the first two objective functions (cost and time) we simply make vector sum for all costs annotating the edges between L vertices. For the third objective function, we count the number of drivers included in the solution, such that successive offers for the same driver counts for one driver.

$$cost(L) = \sum_{i=1}^{|L|-1} c(L_i, L_{i+1}) \tag{1}$$

$$time(L) = \sum_{i=1}^{|L|-1} t(L_i, L_{i+1}) \tag{2}$$

$$drivers(L) = |H|, \forall_{i=1\ldots(|L|-1)} : \tag{3}$$

$$H = \begin{cases} H \mathbin{+\!\!+} \{d(L_i, L_{i+1})\} & \text{if } H_{|H|} \neq d(L_i, L_{i+1}) \\ H & \text{Otherwise} \end{cases}$$

Where H is a list and $\mathbin{+\!\!+}$ is a list concatenation operator.

VI. EXPERIMENTS

To compare the different algorithms in solving the multi-objective route planning problem in a DMR system, a set of experiments have been done and the results of the algorithms are compared in terms of quality and runtime. All experiments are done on a computer with 2G RAM and 2.00GHz CPU running Windows 7. JRE 1.6.0_21 was the runtime environment.

A. Experiment Description

First, we constructed a network of 41 nodes with a density of 0.038 to represent an example road network with nodes number 0 and 41 being at the opposite ends of the network and all other nodes being in-between. Edges are introduced between nodes such that alternative paths between nodes are possible. A time-expanded graph is created by the generation of 500 random trips with length $x, 1 \leq x \leq 13$, on this network.

A random trip is generated by randomly selecting a trip start node, trip length and trip starting time. Trip length represents how many nodes to visit. The trip time and cost between each two nodes are set randomly to represent different behaviors for different drivers and each trip is related for a unique driver. A trip of length x will generate $2 \times x$ vertices in the time-expanded graph as described in section 3. For each algorithm, 100 independent runs were made on the same problem instance.

Algorithms settings are summarized in Table I. Population size and maximum generations were 100 and 250 respectively for all algorithms. For PAES to be equivalent to other algorithms 25000 evaluations were made. All of the algorithms were not very sensitive for the value of the crossover operator. However a value less than 0.4 of the mutation operator results in drop in the solutions' quality. A value more than 0.4 of the mutation operator did not achieve noticeable improvement in the solutions' quality but increases the runtime as it is time consuming.

B. Quality Indicators

Following we describe the metrics used to assess the quality of the approximate Pareto-optimal set $\breve{\Omega}$ generated by each algorithm w.r.t the true Pareto-optimal set Ω generated by GLC. Distances and volumes are measured in the objective space after normalization.

1) Hypervolume: This quality indicator calculates the volume covered by the members of $\breve{\Omega}$ regarding a reference point consisting of the highest values of the objective functions of the elements in Ω. For more information see [15].
2) Generational Distance (GD): This quality indicator is used to measure how far the elements in $\breve{\Omega}$ are from those in Ω as in equation 4.

$$GD = \frac{\sqrt{\sum_{i=1}^{n} {d_i}^2}}{n} \tag{4}$$

where n is the number of the elements in $\breve{\Omega}$, d_i is the Euclidean distance between each element of $\breve{\Omega}$ and the nearest element in Ω. The best value of GD is 0, however this does not necessarily mean $\breve{\Omega} = \Omega$ but means $\breve{\Omega} \subseteq \Omega$. Therefore, we also calculate the Inverted GD.

3) Inverted GD (IGD): Measures how far the elements of Ω are from the elements of $\breve{\Omega}$ as in equation 4.
4) Spread Δ: Defined by Dep et al. [17] for two objectives and generalized by Durillo et al. [19] for more than two objectives. It measures the extent of spread achieved among the elements of $\breve{\Omega}$ as in equation 5.

$$\Delta = \frac{d_f + d_l + d_z + \sum_{i=1}^{n-1} |d_i - \bar{d}|}{d_f + d_l + d_z + (n-1)\bar{d}} \tag{5}$$

Where n is the number of elements in $\breve{\Omega}$, d_i is the Euclidean distance between consecutive elements and $\bar{d}$ is the average of these distances. d_f, d_l and d_z are the

TABLE II
QUALITY INDICATORS. MEAN AND STANDARD DEVIATION

	NSGAII	SPEA2	PESA-II	IBEA	FastPGA	MOCell	PAES
HV	$6.32e{-}01$ $\pm 2.4e{-}02$	$6.28e{-}01$ $\pm 2.4e{-}02$	$6.26e{-}01$ $\pm 2.5e{-}02$	$6.17e{-}01$ $\pm 2.6e{-}02$	$6.25e{-}01$ $\pm 3.0e{-}02$	$6.21e{-}01$ $\pm 2.2e{-}02$	$5.18e{-}01$ $\pm 5.6e{-}02$
GD	$7.76e{-}03$ $\pm 2.3e{-}03$	$8.76e{-}03$ $\pm 3.0e{-}03$	$8.08e{-}03$ $\pm 3.2e{-}03$	$1.09e{-}02$ $\pm 5.1e{-}03$	$9.02e{-}03$ $\pm 4.4e{-}03$	$1.15e{-}02$ $\pm 2.8e{-}03$	$1.49e{-}02$ $\pm 1.1e{-}02$
IGD	$2.11e{-}02$ $\pm 4.9e{-}03$	$2.12e{-}02$ $\pm 4.6e{-}03$	$2.28e{-}02$ $\pm 4.9e{-}03$	$2.71e{-}02$ $\pm 4.4e{-}03$	$2.09e{-}02$ $\pm 4.1e{-}03$	$1.88e{-}02$ $\pm 3.2e{-}03$	$4.49e{-}02$ $\pm 1.5e{-}02$
Δ	$6.93e{-}01$ $\pm 6.4e{-}02$	$6.94e{-}01$ $\pm 6.7e{-}02$	$6.95e{-}01$ $\pm 7.1e{-}02$	$5.12e{-}01$ $\pm 6.7e{-}02$	$6.95e{-}01$ $\pm 5.1e{-}02$	$6.43e{-}01$ $\pm 8.7e{-}02$	$7.60e{-}01$ $\pm 9.9e{-}02$

distances between the border elements in Ω and their nearest elements in $\breve{\Omega}$.

5) Runtime: We compare the actual runtime of the different algorithms.

C. Experimentation Results and Discussion

Table II shows the means and standard deviations for the results obtained regarding the first four quality indicators. Cells marked with dark gray and light gray represent the best and the second best values respectively. We notice that NSGA-II achieved the best results regarding the first quality indicators HV and GD. Having the best values for both of these quality indicators indicates that NSGA-II produces $\breve{\Omega}$ with elements very close ,in objective space, to the elements in Ω. MOCell achieved the best IGD and IBEA achieved the Best Δ.

To determine whether the differences are statistically significant we made the Wilcoxon rank sum test and the results are provided in Tables III-VI. The symbols $\triangledown$, ▲ and – mean row item is worse than column item, row item is better than column item and no significant difference respectively.

Regarding the first metric HV as shown in table III, NSGA-II, SPEA2, PESA-II and FastPGA have no statistically significant difference and PAES has the worst HV.

For the GD metric, PAES, MOCell and IBEA are the worst, second and third worst algorithms respectively as in Table IV. This means that the elements in $\breve{\Omega}$ produced by these algorithms are farther than the elements produced by the other algorithms from the elements in Ω.

Regarding IGD, Table V, MOCell achieved the best results which indicates that the elements produced by MOCell spreads better than all other algorithms as in table VI, except IBEA as will be discussed, making the distance from Ω elements to $\breve{\Omega}$ elements smaller than the other algorithms. However, in objective space, the elements produced by NSGA-II and SPEA2 are closer to the elements in Ω than those produced by MOCell resulting in better HV and GD (better convergence).

IBEA achieved the best Δ as in Table VI. This means that IBEA produces elements that better spreads over the objective space than the other algorithms. Actually Ω itself is not evenly spread therefore Δ is not helpful, in our case, to give indication about the solution quality. However we use it to help explaining other metrics. Although IBEA has the best spread, it suffers from bad IGD as the solutions produced are relatively far from Ω as compared with NSGA-II, SPEA2, PESA-II and FastPGA and does not cover extremes like MOCell as will be discussed.

From Tables III-VI, we notice that all the algorithms that have better HV, GD and IGD than IBEA have worse Δ . This is due to the fact that our Ω is not evenly spread and the algorithms that provide close approximation to this set will not have good Δ. From table II we see that IBEA achieved the second worst HV after PAES.

An interesting point is that IBEA provides better Δ and GD than MOCell but suffers from worse IGD, and both have no statistically significant different HV. IBEA produces elements that are closer to Ω than the elements produced by MOCell which leads to better GD. However, because of the cellular structure of population in MOCell, it allows for better exploration of the search space with high values of the first objective function (time). This makes solutions with high values of time not to be early dominated due to the time-expansion nature of the graph. Therefore MOCell covers some areas in objective space not covered by all of the other algorithms which leads to smaller IGD. Mocell produces better IGD and IBEA produces better GD which leads to statistically no significant difference in HV as in table III.

MOCell achieved the second best Δ, the best IGD and the second worst GD after PAES. However even it has bad GD, still only NGSA-II and SPEA2 provides better HV, thanks for the cellular structure.

In this work, we noticed that Δ is conflicting with GD as Ω is not evenly spread. Also we conclude that the genetic approach outperforms PAES as an evolution strategy which has the worst four metrics values.

Figure 5 shows the results of the runtime comparison with error bars of 3 standard deviations. As discussed in section IV, the major difference between most of the algorithms is the definition and calculation of solution's fitness which plays the major part in the runtime differences. MOCell achieved the best runtime and SPEA2 suffered from the worst runtime. PAES was expected to achieve the best runtime as being the simplest algorithm with only mutation. However, the mutation operator is computationally expensive and executes more in PAES.

Besides the quality of the approximation, runtime is a vital component in the definition of good algorithm in DMR. The competition is mostly between NSGA-II and MOCell which provide good approximation in reasonable time.

VII. CONCLUSIONS AND FUTURE WORK

In this work, we have compared different multiobjective evolutionary algorithms for solving the multiobjective route

TABLE III
WILCOXON TEST: HV

	SPEA2	PESA-II	IBEA	FastPGA	MOCell	PAES
NSGAII	–	–	▲	–	▲	▲
SPEA2		–	▲	–	▲	▲
PESA-II			▲	–	–	▲
IBEA				▽	–	▲
FastPGA					–	▲
MOCell						▲

TABLE IV
WILCOXON TEST: GD

	SPEA2	PESA-II	IBEA	FastPGA	MOCell	PAES
NSGAII	▲	–	▲	▲	▲	▲
SPEA2		–	▲	–	▲	▲
PESA-II			▲	–	▲	▲
IBEA				▽	▲	▲
FastPGA					▲	▲
MOCell						▲

TABLE V
WILCOXON TEST: IGD

	SPEA2	PESA-II	IBEA	FastPGA	MOCell	PAES
NSGAII	–	▲	▲	–	▽	▲
SPEA2		▲	▲	–	▽	▲
PESA-II			▲	▽	▽	▲
IBEA				▽	▽	▲
FastPGA					▽	▲
MOCell						▲

TABLE VI
WILCOXON TEST:Δ

	SPEA2	PESA-II	IBEA	FastPGA	MOCell	PAES
NSGAII	–	–	▽	–	▽	▲
SPEA2		–	▽	–	▽	▲
PESA-II			▽	–	▽	▲
IBEA				▲	▲	▲
FastPGA					▽	▲
MOCell						▲

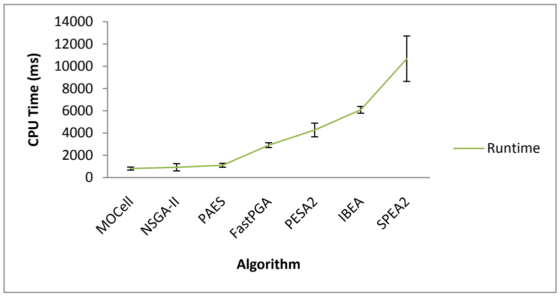

Fig. 5. Runtime comparison

planning problem modeled as a time-expanded graph. The set of algorithms includes six genetic multiobjective evolutionary algorithms in addition to one evolution stragey algorithm. The evolution strategy showed to be the worst for solving our problem. Among the six genetic algorithms no single algorithm wears the crown for our problem but a subset includes NSGA-II and MOCell. NSGA-II converges better than MOCell but the latter covers the objective space better in less time. The time-expanded model of the problem pushed the algorithms to a local optima where the search is concentrated in the higher parts of the graph which has lower values of the time objective. The cellular structure of MOCell helped it in resisting this problem as the population is distributed in cells and the selection of parents in each cell is independent of the other cells. In future work, we consider taking advantage of the strengths of each algorithm to combine them and avoid the weaknesses. We consider combining the good NSGA-II convergence and the good spread and diversity of MOCell.

ACKNOWLEDGMENT

This work was partially supported by the German Academic Exchange Service (DAAD), Grant no. A/08/99166. The authors would like to thank the anonymous reviewers for their valuable comments.

REFERENCES

[1] M. Mann, R. Bradley, and M. Hughes, “Northern hemisphere temperatures during the past millennium: Inferences, uncertainties, and limitations,” *GRL*, vol. 26, pp. 759–762, 1999.

[2] D. Sperling and J. Cannon, “Climate and transportation solutions: Findings from the 2009 asilomar conf. on trans. and energy policy,” Uni. of California, Inst. of Trans. Studies, Tech. Rep., 2010.

[3] EPA, “Greenhouse gas emissions from the U.S. transportation sector, 1990-2003,” U.S. EPA, Office of Transportation and Air Quality, Tech. Rep., 2006.

[4] (2010) Occupancy rates of passenger vehicles (term 029). European Environment Agency. [Online]. Available: http://www.eea.europa.eu/data-and-maps/indicators/occupancy-rates-of-passenger-vehicles/occupancy-rates-of-passenger-vehicles-1

[5] (2009) 2009 national household travel survey. U.S. Department of Transportation, Federal Highway Administration. [Online]. Available: http://nhts.ornl.gov

[6] D. Schrank and T. Lomax, “The 2007 urban mobility report,” Texas Transportation Inst., Washington, Tech. Rep., 2007.

[7] S. Jacobson and D. King, “Fuel saving and ridesharing in the U.S.: Motivations, limitations, and opportunities,” *Transportation Research Part D: Trans. and Environment*, vol. 14, no. 1, pp. 14 – 21, 2009.

[8] H. S. and M. Buchmann, “Empty seats traveling,” Nokia Research Center, Bochum, Tech. Rep., 2007.

[9] N. Agatz, A. Erera, M. Savelsbergh, and X. Wang, “Sustainable passenger transportation: Dynamic ride-sharing,” Erasmus Research Inst. of Management (ERIM), Erasmus Uni., Rotterdam, Tech. Rep., 2010.

[10] P. Gruebele, “Interactive system for real time dynamic multi-hop carpooling,” Global Transport Knowledge Partnership, Tech. Rep., 2008.

[11] E. Zitzler, L. Thiele, M. Laumanns, C. Fonseca, and V. da Fonseca, “Performance assessment of multiobjective optimizers: an analysis and review,” *IEEE TEC*, vol. 7, no. 2, pp. 117 – 132, apr. 2003.

[12] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. NY, USA: W. H. Freeman & Co., 1979.

[13] A. Skriver and K. Andersen, “A label correcting approach for solving bicriterion shortest-path problems,” *Computers & Operations Research*, vol. 27, no. 6, pp. 507 – 524, 2000.

[14] A. Abraham, L. Jain, and R. Goldberg, *Evolutionary Multiobjective Optimization: Theoretical Advances and Applications (Advanced Information and Knowledge Processing)*. NJ, USA: Springer-Verlag New York, Inc., 2005.

[15] A. Nebro, J. Durillo, F. Luna, B. Dorronsoro, and E. Alba, "Mocell: A cellular genetic algorithm for multiobjective optimization," *IJIS*, vol. 24, no. 7, pp. 726–746, 2009.

[16] H. Eskandari, C. Geiger, and G. Lamont, "FastPGA: A dynamic population sizing approach for solving expensive multiobjective optimization problems," in *EMO 2007*, ser. LNCS, S. Obayashi, K. Deb, C. Poloni, T. Hiroyasu, and T. Murata, Eds., vol. 4403. Springer, 2007, pp. 141–155.

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

[18] J. Knowles and D. Corne, "The pareto archived evolution strategy: A new baseline algorithm for multiobjective optimization," in *Proceedings of the 1999 Congress on Evolutionary Computation*. Piscataway, NJ: IEEE Press, 1999, pp. 9–105.

[19] J. Durillo, A. Nebro, and E. Alba, "The jmetal framework for multi-objective optimization: Design and architecture," in *IEEE CEC*, Barcelona, Spain, July 2010, pp. 4138–4325.

[20] F. Schulz, D. Wagner, and K. Weihe, "Dijkstras algorithm on-line: An empirical case study from public railroad transport," in *Alg. Eng.*, ser. LNCS, J. Vitter and C. Zaroliagis, Eds. Springer Berlin / Heidelberg, 1999, vol. 1668, pp. 110–123.

[21] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the strength pareto evolutionary algorithm for multiobjective optimization," in *Evolutionary Methods for Design Optimization and Control with Applications to Industrial Problems*, K. C. Giannakoglou, D. T. Tsahalis, J. Périaux, K. D. Papailiou, and T. Fogarty, Eds. Athens, Greece: International Center for Numerical Methods in Engineering, 2001, pp. 95–100.

[22] D. W. Corne, N. R. Jerram, J. D. Knowles, M. J. Oates, and M. J, "PESA-II: Region-based selection in evolutionary multiobjective optimization," in *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2001)*. Morgan Kaufmann Publishers, 2001, pp. 283–290.

[23] E. Zitzler and S. Knzli, "Indicator-based selection in multiobjective search," in *Parallel Problem Solving from Nature - PPSN VIII*, ser. Lecture Notes in Computer Science, X. Yao, E. Burke, J. A. Lozano, J. Smith, J. J. Merelo-Guervs, J. A. Bullinaria, J. Rowe, P. Tino, A. Kabn, and H.-P. Schwefel, Eds. Springer Berlin / Heidelberg, 2004, vol. 3242, pp. 832–842.

[24] D. Costelloe, P. Mooney, and A. Winstanley, "From random walks to pareto optimal paths," in *4th Irish AI and Cog.Science*, m, Ed., 2001, pp. 309 – 318.

[25] P. Mooney and A. Winstanley, "An evolutionary algorithm for multicriteria path optimization problems," *IJGIS*, vol. 20, pp. 401 – 423, 2006.