A Multiobjective Memetic Algorithm for Solving the Carsharing Problem

Laurent Moalic, Sid Lamrous, Alexandre Caminada

A Multiobjective Memetic Algorithm for Solving the Carsharing Problem

Laurent Moalic, Sid Lamrous, and Alexandre Caminada
IRTES-SET, Belfort-Montbliard University of Technology - UTBM
F-90010 Belfort, France
{laurent.moalic, sid.lamrous, alexandre.caminada}@utbm.fr

Abstract— *In this article, we present a method combining a genetic approach with a local search for multiobjective problems. It is an extension of algorithms for the single objective case, with specific mechanisms used to build the Pareto set. The performance of the proposed algorithm is illustrated by experimental results based on a real problem with three objectives. The problem is issued from electric carsharing service with a car manufacturer partner. Compared to the Multiobjective Pareto Local Search well known in the scientific literature, the proposed model aims to improve: the solutions quality and the set diversity.*

Keywords: Memetic algorithm, Local search, Multiobjective optimization, Transportation services, Carsharing, Decision making

1. Introduction

Many real world problems require to optimize several objectives simultaneously, they are called multiobjective optimization problems (MOP).

The general MOP can be formulated as:

$$MOP \begin{cases} \text{max} & z = f(x) \\ \text{where} & f(x) = (f_1(x), f_2(x), ..., f_k(x)) \\ \text{subject to:} & x \in X \end{cases} \tag{1}$$

In most cases considering a multiobjective context, it does not exist a unique solution optimizing all objectives in an optimal way. We need to find other decisional mechanisms. The Pareto dominance is one of these; for MOP, the Pareto set is composed of all best compromises between the different objectives. The Pareto set is achieved if there are no other dominant solutions in the search space. The Pareto front is defined as the image of the Pareto set in the objective space [1]. The Vector Evaluated Genetic Algorithm proposed in 1985 [2] was at the origin of many works using genetic approaches (GA) for solving MOP called multiobjective evolutionary algorithms (MOEA). Some famous algorithms such NSGA-II[3], SPEA [4] and SPEA2 [5] have shown their efficiency on complex problems. To solve single objective combinatorial optimization problems, local search algorithms provide often efficient metaheuristics. They have also been adapted to multiobjective combinatorial problems like in Pareto Local Search algorithm (PLS) [6] or more recently in FLSMO [7]. Some other works use strategies based on the neighborhood structure [8] or consider a Tabu Search approach [9][10] . A recent work has been done to unify all local search approaches based on a neighborhood search applied to MOP. This unification is called Dominance-based Multiobjective Local Search (DMLS) [11]. Both local search and GA bring interesting behavior. It is why in the past few years, many multiobjective algorithms have proposed to hybridize GA with local search. These approaches are frequently called Memetic Algorithms or Hybrid Genetic Algorithm. Such algorithms have often outperformed results obtained with simple local search or simple GA, especially in a mono-objective context such for the graph coloring problem [12]. In a multiobjective context, memetic algorithms seems to be very promising [13][14] or [15].

The algorithm we propose is an hybridization of a genetic algorithm based on NSGA-II and a local search. Hybridizing NSGA-II and a local search has been done in some works like in [15]. But in our algorithm the local search is not performed after the GA but inside it, instead of the mutation operator. For each generation of the GA, children are obtained by a crossover mechanism from the selected parents and a local search is applied to the offsprings instead of the mutation. By this way the crossover can be considered as the diversification operator while the local search is the intensification one. A good balance between intensification and diversification allows the algorithm to approach the Pareto set.

The proposed algorithm has been applied on a real problem for locating stations of a Carsharing service for electrical vehicles. The good results we obtained in comparison to PLS shows how much the hybridization is one of the most promising directions of research.

The article is structured as follows. In Section 2, we present the proposed memetic algorithm. Section 3 proposes a detailed presentation of the experimentation performed in Paris area, where the hybrid algorithm was applied on a Carsharing service for electrical vehicles. Then we present in section 4 the decision making tool we develop. Section 5 provides an analysis of the experimental results. The last section presents a conclusion and considers future directions.

2. Memetic algorithm for Multiobjective problems

The proposed algorithm is based on an hybridization of two main approaches: a GA and a local search. It is a population-based algorithm which manages two population sets. The first population with a fixed size is used by the GA for selecting promising parents. The second population is an elitist set used as an archive for collecting all non-dominated solutions reached by the algorithm. During the local search, each acceptable neighbor is stored in the archive if it is not dominated by the solutions found so far.

The GA used in the hybridization is NSGA-II[3] proposed by Deb et al. This algorithm is characterized by:

- a fast non-dominated sorting approach according to the level of non-domination
- a density estimation of solutions surrounding known as the crowding distance
- a sorting function according firstly to the level of non-domination and secondly to the crowding distance

Let's remind that the level of non-domination is computed as fallows:

1) All non-dominated solutions have the level 1
2) The non-dominated solutions of level 1are removed
3) All remaining non-dominated solutions have level 2
4) The non-dominated solutions of level 2 are removed
5) So on...

The crowding distance for a solution s is defined as the average side-length of the cuboid defined with the previous and the next closest solutions.

The algorithm 1 presents these steps.

Algorithm 1 Main Memetic: NSGA-II based approach

Require: population size N, generation number $nbIter$ $\{nbIter$ can be replaced by a time limit$\}$

$P \leftarrow$ init(N) {init population P with N random individuals}
$Q \leftarrow \emptyset$ {init an empty children population}
eval(P) {eval objectives for each individual}
for $i = 1$ to $nbIter$ **do**
  $P \leftarrow P \cup Q$
  assignRank(P) {based on Pareto dominance}
  for each non-dominated front $f \in P$ **do**
    setCrowdingDist(f)
  end for
  sort(P) {by rank and in each rank by the crowding dist }
  $P \leftarrow P[0 : N]$
  $Q \leftarrow$ buildChildren(P)
end for

Moreover NSGA-II as a GA needs operators for selection/crossover of the parents and mutation. Since here we consider an hybrid algorithm, mutation will be replaced by the local search in a selected direction. These important steps are done by the function called buildChildren(P) as follows:

- selection: as defined in NSGA-II this is done by a binary tournament based on the solutions ranking
- choice of a random direction combining the objective functions
- crossover: an elitist recombination for the chosen direction
- mutation: child improvement with a local search in the chosen direction

The initial population is built randomly, with n random solutions. The result is an approximation of the Pareto set. The proposed approach combines two qualities: a good intensification based on the local search and a good diversification thanks to the crossover.

2.1 Random choice of the direction to explore

At the previous step solutions were ranked according to the Pareto dominance. Now for crossover and local search a direction $\omega = (\omega_1, ..., \omega_k)$ is randomly chosen with k the number of objective functions, $\omega_i \geq 0$ and $\sum_{i=1}^{k} \omega_i = 1$.

The vector direction allows to transform temporarily the MOP to a single-objective problem. Each solution has a unique fitness value f which is the weighted sum of objectives values f_i.

$$f = \sum_{i=1}^{k} \omega_i * f_i \tag{2}$$

This allows to get an ordered relationship between solutions.

2.2 Crossover in the GA

In the memetic approach the crossover may be seen as a diversification operator. In our algorithm we propose to control the diversification with an elitist mechanism.

We suppose here that each variable of an individual is associated to its contribution to every objective functions. The crossover will select from each parent the best variable value according to the selected direction.

2.3 Local Search instead of the mutation

For this operator we propose to use a fast local search algorithm. The selected one is First Improvement Hill Climbing (FIHC) in which the first neighbor with a better quality is chosen (partial neighborhood exploration). Local search is a pair (Ω, V) where Ω is a set of feasible solutions (search space) and V a neighborhood structure $V : \Omega \rightarrow 2^{\Omega}$ that assigns to every $s \in \Omega$ a set of neighbors $V(s)$. Local search is applied until being in a local optimum s^* such that $\forall s \in V(s^*), f(s) \leq f(s^*)$ for a maximization problem.

Algorithm 2 Local Search: FIHC

Require: s the child coming from crossover operator
$f_s \leftarrow \sum_{i=1}^{k} \omega_i * f_i(s)$
repeat
  $s' \leftarrow$ selectNeighbor(s) {select randomly the first neighbor of s such that $f'_s > f_s$}
  if $s' \neq \varnothing$ **then**
    $s \leftarrow s'$
    $f_s \leftarrow \sum_{i=1}^{k} \omega_i * f_i(s)$
    addNotDominated(s) {add s in the archive if not dominated and remove all dominated solutions}
  end if
until s is a local optimum (i.e. $s' = \varnothing$)

3. Study case: Problem and data

In our study we consider a Carsharing system. Carsharing services were first experimented in 1940 [16]. To deploy the service, we need to locate stations where the people take and return the cars. Here it is not necessary to return the vehicle in its starting station. This approach where vehicles can be return to any station of the service is known as One Way Carsharing. Solving approaches based on exact methods already exist such as [17] but they consider simplified problem. We have applied the proposed memetic algorithm to approximate the Pareto set of this problem. The aim is to locate n stations in a given area to maximize the daily requests of population flows.

As it will be detailed in section 3.3, our model aims to:

- increase the number of people using the service
- reduce the jockeying effort (relocation operation performed by a staff) to alleviate the number of rejected demands
- increase the use of the service during all the day (not only at peak time)

A client's demand is considered as rejected in two cases:

- the client arrives to a station to rent a car and couldn't find any available one
- the client arrives to a station to park the car and couldn't find an empty place

3.1 Mobility model

In this section we will describe the mobility model used in our study and how we used it to generate the input data for our heuristic.

To build the mobility model, several data are necessary to simulate the real environment and people mobility. We used two main types of data:

- GIS Shapefiles describing the geographical entities.
- Survey data and socio-economical information collected by professionals for regional planning needs, describing the main flows of people mobility.

For our study the data are on a 20km * 10km area in Paris region, France. The area is divided into a grid of equal cell size. Each cell is characterized by two properties:

- Terrain type: static information representing the dominant structure type of the area covered by the cell (roads, buildings, houses, company, etc.).
- Attraction weight: dynamic information that varies every 15 minutes during the day. Attraction weights are computed according to cells terrain type and all other survey data.

A model that we have developed before is used to build the people mobility between each cell during the day. All the flows are set in a 3D matrix $F = (f_{i,j,t})$ where $f_{i,j,t}$ represents the number of people moving from the cell i to cell j at time period t.

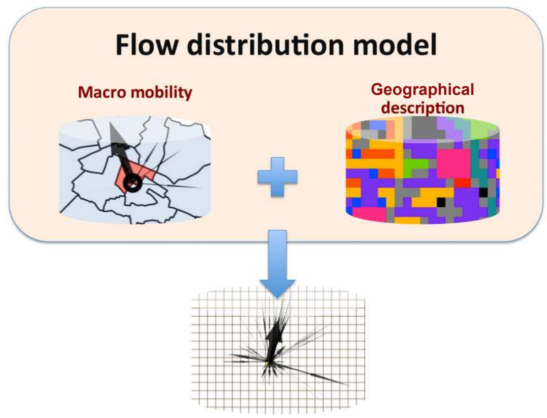

Fig. 1
FLOWS DISTRIBUTION IN CELL

The figure 1 shows the outflows from the selected cell, obtained from macro data and geographical description. The thickness of the arrows reflects the number of people moving between cells. For the purpose of the study we define a cells size of 300m side. Not all the cells can be the origin or destination of a mobility flows such as fields or lakes. For our experimentation nearly 400 cells are considered. That gives 160,000 origin/destination couples. Considering time slots of ¼ hour (96 per day), the flow matrix used contains more than 15,000,000 records describing how people are moving during the day.

3.2 Forecasting on the service users

The algorithm for locating the stations and the evaluation of users' number for each time suppose a good understanding of who will use the service. This part was achieved with the operator who will deploy the Carsharing service.

Among all the mobility flows, some filters are used for selecting the rate of users depending on their profile (age, sex, etc.). Moreover a station is defined by a capture radius which defines the maximum distance the users are ready to walk to reach the station. Based on this radius value, a station st_i can cover one or more cells. In territory planning it is generally considered that one accept to walk 300m to join a public transportation station.

Fig. 2
STATIONS AND CELLS COVERAGE

In figure 2 are shown stations and their coverage area (colored disc). The inflows / outflows associated to each station are the weighted sum of the flows from the covered cells. The weight depends on the proportion of coverage. If several stations cover the same part of a cell, the associated flows are divided between all of them.

3.3 Stations location

In this section we will discuss how to locate the car stations in the zone of study such that they cover the maximum demand in an appropriate manner.

Stations are located by the algorithm by optimizing three objectives:

f1 : flow maximization i.e. the locations must allow us to maximize the flows between themselves

$$f_1 = \max_{s\in\Omega}\left[\sum_{st_i\in s}\sum_{st_j\in s\setminus\{st_i\}} f(st_i, st_j)\right] \quad (3)$$

f2 : balance maximization i.e. the location must allow us to maximize the balance between inflows and outflows of a station

$$f_2 = \max_{s\in\Omega}\left[\sum_{st_i\in s}\frac{f_r(st_i)}{f_T(st_i)}\right] \quad (4)$$

f3 : minimization of flow standard deviation i.e. the location must allow us to get an uniform flow along the day

$$f_3 = \min_{s\in\Omega}\left[\sum_{st_i\in s}\sqrt{\frac{1}{|T|}\sum_t (f(st_i,t) - \bar{f}(st_i))^2}\right] \quad (5)$$

With,

Ω : set of feasible solutions

s : solution element of Ω corresponding to a network of n charging stations

st_i : charging station i from the solution s

T : set of time periods of the day

t : one time period (for instance 15 minutes)

$f(st_i, st_j)$: number of people moving from st_i to st_j on all time periods

$f(st_i, st_j, t)$: number of people moving from st_i to st_j on time period t

$f(st_i, t)$: number of people moving from/to st_i on time period t

$\bar{f}(st_i)$: average number of people moving from/to st_i on all time periods

$f_r(st_i) = \sum_t \min\left[\sum_{st_j\in s\setminus\{st_i\}} f(st_i, st_j, t), \sum_{st_j\in s\setminus\{st_i\}} f(st_j, st_i, t)\right]$ is the balanced part of the in/out flow throughout the day

$f_T(st_i) = \sum_t \max\left[\sum_{st_j\in s\setminus\{st_i\}} f(st_i, st_j, t), \sum_{st_j\in s\setminus\{st_i\}} f(st_j, st_i, t)\right]$ is the total flow going through st_i station

The neighborhood of a solution s, needed for the local search, is composed of all the solutions that can be reached by moving any one station of s to any other location. The neighborhood relation is then defined as:

$$v(s) = \{s' \in \Omega | dist(s, s') = 1\} \quad (6)$$

4. Simulation tool and platform of analysis

Good mobility analysis requires an effective tool for representing the results of displacement. A software platform that can run both spatial data such as geographical and time data reflecting the mobility then appears as an essential element.

Fig. 3
GEOLOGIC PLATFORM

Many tools exist in the area of GIS (Geographic Information Systems) to represent geo-located data. However, taking into account both space and time is scarce.

So we have developed our own software platform named ”GeoLogic”. This tool proposes all features found in traditional GIS, such as the display of maps and ”shapefiles” vector data, a zoom, etc. Each loaded object, such as roads or buildings delimitations, is a layer that can be shown or hidden.

Moreover this tool reveals in the same view the decision space (geographical part) and the criteria space (plot part). Associated tools allow a decision maker to view the evaluation of a solution, and for a selected location the associated evaluations.

Fig. 4
VECTOR INFORMATION: GIS SHAPEFILE

The data can be static or dynamic. While topographic features such as roads or buildings do not vary over time, their properties can change during the day. As an example a supermarket will always be in the same place but will have a different power of attraction at 9:00, 16:00 or 23:00. Similarly the location of the population changes over time. Our software allows taking into account this dynamicity over time. It is possible to vary the speed of scrolling or to pause the application.

5. Performance analysis

This section deals with the performance measurement and the results. In the context of multiobjective optimization the comparison of algorithms is quite more difficult than in single-objective case. Indeed the result is not only one solution but a set of solutions, and each solution has an evaluation on several axes. As long as there is no fitness value associated to the population one can’t say which population is better than the other.

In the general case for two approximations of the Pareto front one front can be better for a criteria but worst for another one. As an example figure 5 shows two uncomparable sets since they cannot be totally ordered. The only situation in which a direct comparison is possible is when a set is totally dominated by another one. Then we will use specific indicators explained in the following section.

5.1 Results analysis

For analyzing the results we performed 20 runs of the memetic algorithm, and we will compare the results to a

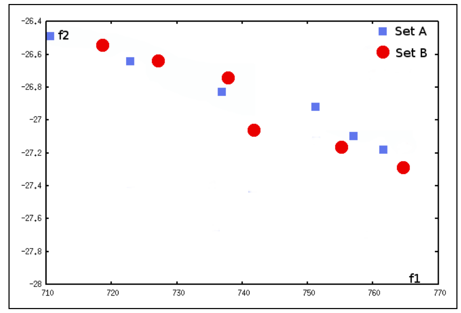

Fig. 5
A MULTIOBJECTIVE MAXIMISATION PROBLEM

reference set obtained after 20 runs of PLS. The average time before being in a local optimum for PLS is 2 hours. It is the computation time we have chosen for each run of our algorithm. By this way one can consider that the results of both algorithm are comparable.

Figures 6, 7 and 8 presents the projection of the non-dominated solutions for a typical run. Each color/shape presents the solutions for our memetic algorithm and the local search PLS.

The first important result we can see is that the memetic approach produces solutions in zones of the criteria space which are never explored by PLS.

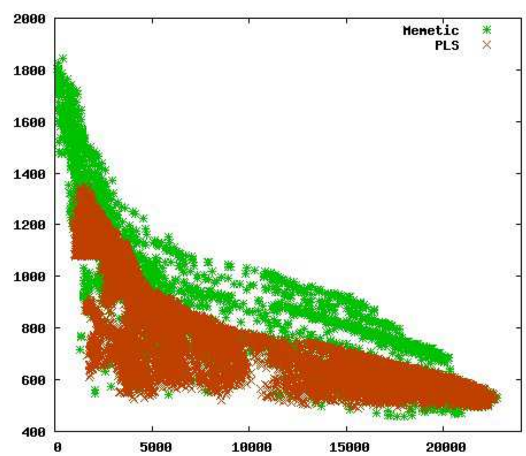

Fig. 6
PROJECTION ON F1 (X) AND F2 (Y) - MAXIMIZING

For a more precise analyze of the results we will use indicators. Indeed a comparative indicator is the most common way to distinguish two sets or to assign a performance measure. Since an indicator gives a restrictive information we propose for this work to consider three indicators:

- the additive ϵ-indicator [18]
- the contribution indicator [19]
- the number of solutions found

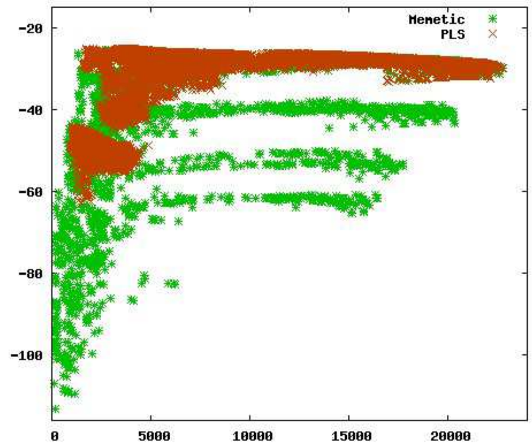

Fig. 7
PROJECTION ON F1 (X) AND F3 (Y) - MAXIMIZING

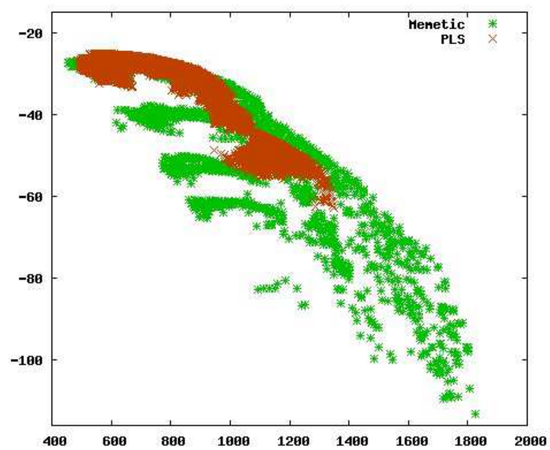

Fig. 8
PROJECTION ON F2 (X) AND F3 (Y) - MAXIMIZING

additive ϵ-indicator

The additive ϵ-indicator gives the minimum factor ϵ by which a set A has to be translated to dominate the set B.

$$I_{\epsilon+}(A,B) = \min_{\epsilon \in \mathbb{R}} \{\forall x \in B, \exists x' \in A : x \preceq_{\epsilon+} x'\} \quad (7)$$

In the equation 7 the $\preceq_{\epsilon+}$ operator represents the weak ϵ-dominance. For a maximization problem we have the relation:

$$x \preceq_{\epsilon+} x' \Leftrightarrow \forall i \in \{1,2,...,n\}, x_i \leq \epsilon + x'_i$$

$I_{\epsilon+}$ is sensitive to the dimension of each objective function. Before comparing solutions in any computation, a normalization of the axes is done.

Table 1
MEMETIC RESULTS

	min	max	mean
f_1	22787.801	22787.801	22787.801
f_2	1783.280	1874.630	1826.831
f_3	-25.217	-25.047	-25.150
NbSol	4515	4915	4766
$\epsilon+$	5.135	2.921	3.719
contrib	0.104	0.117	0.109

Table 2
PLS RESULTS

	min	max	mean
f_1	22787.801	22787.801	22787.801
f_2	1024.540	1347.700	1076.144
f_3	-25.042	-25.039	-25.041
NbSol	8249	12969	9094
$\epsilon+$	76.621	17.232	66.050
contrib	0.202	0.300	0.226

contribution indicator

The contribution indicator computes the proportion of solutions from a set A in $ND(A \cup B)$, where ND represents the non dominated solutions.

$$I_C(A,B) = \frac{\frac{|A \cap B|}{2} + |W_A| + |N_A|}{|ND(A \cup B)|} \quad (8)$$

The indicators $I_{\epsilon+}$ and I_C are used to compare the results of algorithms to the same reference set R. In an ideal situation R would be the optimal Pareto set. For our real problem the optimal Pareto set is unknown. It is why we have built R with all non-dominated solutions found on many runs of PLS and FLSMO.

An $I_{\epsilon+}$ value near to 0 or being negative shows a very good result. In the same way the bigger I_C value is, the better the set is.

The results presented in these tables 1 and 2 give information on the fitnesses evaluation and the three indicators. For each we indicate the best value on the 20 runs, the worst of the max values, and the mean.

A significant result is that our memetic approach can't find as many solutions as PLS. But the found solutions are better spread in the criteria space. The evaluation of f_2 confirms that the results are significantly better with the memetic algorithm than with PLS.

5.2 Progression of the algorithm

Moreover we propose to study how the archive increase during the program run. This allows to compare the way the algorithms converge during the process. To do that we consider the ϵ-indicator evolution. After each offspring has been improved by the local search, the archive is compared to the reference set. By this way it is possible to compare not only the results of an algorithm but the way to reach these results.

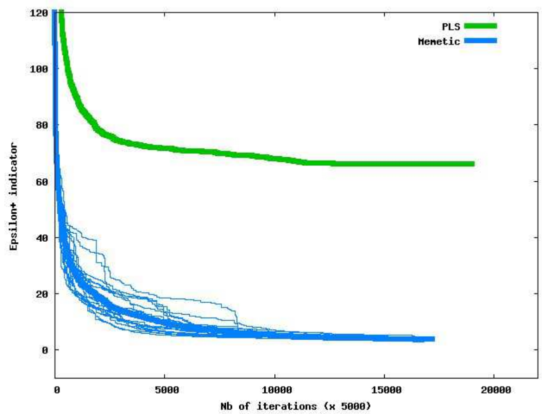

Fig. 9

ϵ-INDICATOR EVOLUTION FOR THE MEMETIC AND THE REFERENCE PLS

Figure 9 shows the comparison on 20 runs between our Memetic algorithm and the reference algorithm PLS. The results given by the Memetic algorithm seem to be very promising. The curves revel that after only few minutes of computation the hybrid algorithm give better results than PLS after 2 hours according to the selected indicator.

6. Conclusion and future works

We have proposed in this article three main contributions. The first one is about a memetic approach for solving multiobjective problems. This algorithm hybridizes a GA based on NSGA-II and a local search like FIHC. The second contribution is a new way for modeling the problem of locating stations for a Carsharing service. This model is based on a dynamic knowledge of how people move on the territory during the day. To solve this problem in a real context we have proposed three objectives to optimize:

- to increase the number of people using the service
- to reduce the jockeying effort to alleviate the number of rejected demands
- to increase the use of the service all during the day

The last main contribution is about a decision making tool which shows in a same view the decision and criteria spaces. This interactive tool allows the users to understand interactions between the solutions and their evaluation.

The evaluation of our algorithm was compared to the PLS algorithm results for three indicators. The main important aspect is that even if PLS provides nearly twice more non-dominated results, these solutions are in a small part of the criteria space. The algorithm we proposed provides a very good diversification and intensification, in a comparative computation time.

These results confirms that hybridizing GA and local search is a very promising way for future researches.

As a perspective it seems interesting to apply a Pareto based local search such as PLS or FLSMO on the archive set at the end of the memetic algorithm. This new hybridization could complete the archive by finding new solutions in the neighborhood of the non-dominated solutions.

References

[1] C. Coello and G. Lamont, *Applications of multi-objective evolutionary algorithms*, vol. 1. World Scientific Publishing Company Incorporated, 2004.

[2] J. D. Schaffer, "Multiple objective optimization with vector evaluated genetic algorithms," in *Proceedings of the 1st International Conference on Genetic Algorithms*, (Hillsdale, NJ, USA), pp. 93–100, L. Erlbaum Associates Inc., 1985.

[3] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: Nsga-2," *Evolutionary Computation, IEEE Transactions on*, vol. 6, pp. 182 –197, apr 2002.

[4] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach," *Evolutionary Computation, IEEE Transactions on*, vol. 3, pp. 257 –271, nov 1999.

[5] E. Zitzler, M. Laumanns, and L. Thiele, "Spea2: Improving the strength pareto evolutionary algorithm," *TIK-Report 103*, 2001.

[6] L. Paquete, M. Chiarandini, and T. Stützle, "Pareto local optimum sets in the biobjective traveling salesman problem: An experimental study," in *Metaheuristics for Multiobjective Optimisation* (X. Gandibleux, M. Sevaux, K. Sörensen, V. T'kindt, G. Fandel, and W. Trockel, eds.), vol. 535 of *Lecture Notes in Economics and Mathematical Systems*, pp. 177–199, Springer Berlin Heidelberg, 2004.

[7] L. Moalic, A. Caminada, and S. Lamrous, "A fast local search approach for multiobjective problems," in *LION*, (Catania, Italie), january 2013.

[8] Z. Wu and T. S. Chow, "A local multiobjective optimization algorithm using neighborhood field," *Structural and Multidisciplinary Optimization*, pp. 1–18, 2012.

[9] X. Gandibleux and A. Freville, "Tabu search based procedure for solving the 0-1 multiobjective knapsack problem: The two objectives case," *Journal of Heuristics*, vol. 6, pp. 361–383, 2000. 10.1023/A:1009682532542.

[10] M. P. Hansen, "Tabu search for multiobjective optimization: Mots," in *MCDM'97*, Springer-Verlag, 1997.

[11] A. Liefooghe, J. Humeau, S. Mesmoudi, L. Jourdan, and E.-G. Talbi, "On dominance-based multiobjective local search: design, implementation and experimental analysis on scheduling and traveling salesman problems," *Journal of Heuristics*, vol. 18, pp. 317–352, 2012. 10.1007/s10732-011-9181-3.

[12] P. Galinier and J.-K. Hao, "Hybrid evolutionary algorithms for graph coloring," *Journal of Combinatorial Optimization*, vol. 3, no. 4, pp. 379–397, 1999.

[13] J. Knowles and D. Corne, "M-paes: a memetic algorithm for multiobjective optimization," in *Evolutionary Computation, 2000. Proceedings of the 2000 Congress on*, vol. 1, pp. 325 –332 vol.1, 2000.

[14] A. Jaszkiewicz, "Genetic local search for multi-objective combinatorial optimization," *European Journal of Operational Research*, vol. 137, no. 1, pp. 50 – 71, 2002.

[15] K. Deb and T. Goel, "Controlled elitist non-dominated sorting genetic algorithms for better convergence," in *Proceedings of the First International Conference on Evolutionary Multi-Criterion Optimization*, EMO '01, (London, UK, UK), pp. 67–81, Springer-Verlag, 2001.

[16] S. A. Shaheen and A. P. Cohen, "Worldwide carsharing growth: An international comparison," 2008.

[17] G. H. de Almeida Correia and A. P. Antunes, "Optimization approach to depot location and trip selection in one-way carsharing systems," *Transportation Research Part E: Logistics and Transportation Review*, vol. 48, no. 1, pp. 233 – 247, 2012.

[18] E. Zitzler, L. Thiele, M. Laumanns, C. Fonseca, and V. da Fonseca, "Performance assessment of multiobjective optimizers: an analysis and review," *Evolutionary Computation, IEEE Transactions on*, vol. 7, pp. 117 – 132, april 2003.

[19] H. Meunier, E.-G. Talbi, and P. Reininger, "A multiobjective genetic algorithm for radio network optimization," in *In Proceedings of the 2000 Congress on Evolutionary Computation*, pp. 317–324, IEEE Press, 2000.