Touristic guide for city guidance with minimum-maximum optimization multi-criteria model: Budget VS Number of cities

Guzmán S. Author and Manrique J. Author

Abstract – **This articles consists in developing a description of the implementation of a modelling and optimization process on the development of a software tool for trip guidance among cities with the goal to maximize the quantity of visited places and minimizing the budget spent on the trip. With the use of a mathematical model, verifying with a computational methods tool (GAMS) and generating appropriate metaheuristics (SPEA 2), this paper illustrates the basis for implementing a project of such magnitude and what it would take to make it a reality.**

Key Terms – **Multi-objective optimization, evolutionary algorithms, mathematical modelling, touristic guidance, transport.**

I. INTRODUCTION

TRAVELLING represents one of the most enriching experiences for the contemporaneous human because it allows him to leaving his sedentary nature and discover new places, flavors and people. In spite of this, not everyone has the same preferences or restrictions at the moment of planning a new journey, in other words, parameters as the number of available days for travelling, cities for visiting, budget, among others, are too specific for each traveler. This means that there exist numerous ways to execute a journey, for which it is desirable that every planned journey is the best among the available alternatives.

Talking about the state of the art for this particular problem, there are some applications can be classified among the following types:

- Plan routes with the least cost distance between points: In this regard, applications like Rome2Rio and Google Maps can be taken into account. Among various functionalities Google Maps offers, the most famous ones consist on showing detailed information about geographical regions and sites worldwide with the “best” route (the one with lowest costs of transportation) that allows you to go from one place to another [5]. In the case of Rome2Rio, it offers the possibility to query for any route between cities cities, landmarks, attractions or addresses across the globe in a multi-modal manner. [4]
- Plan activities to do in a particular city: For this category, the best solution available is “Google Trips”, which is a mobile application that gathers your travel information from Gmail and organizes daily itineraries for your time being in the specified city (the app can work without the use of internet) [1]. Also, other solutions like “Sygic Travel” just receive the place you are going to and the number of days as an input in order to perform your itinerary during the whole trip. [3].

As it can be seen, the specific problem that is being treated inside this article is not formally solved by any of the existing tools, that either generate routes with the lowest transportation costs for going from one city to another, or create a daily itinerary with all the activities that the person can do during the whole trip.

II. PROBLEM DEFINITION

From the point of view of the traveler, the best journey is that which minimizes costs. Here, a good journey is defined by the quantity of visited places and the quality of these. In other words, the problem to attack is to minimize the cost of travelling maximizing the amount and quality of the visited places. Under this premise, the variables taken into consideration re the next ones:

Places to visit: Each traveler has a list of the places he would like to potentially visit in his next journey.

Places’ score: Each place must have a score which must be comparable to other places´, in order to decide which of them is better.

Time: A traveler has a limited time window to accomplish his journey.

Living cost (daily): The traveler must afford his own living in each place he visits. For this, the cost of average living per place is going to be used.

Transport costs: Moving from one place to another implies some costs. For now, only terrestrial means will be considered in such parameter.

Maximum number of days per place: This, with the goal of giving every traveler enough time (the one he/she considers necessary) to know each place he visits, meaning, the planned number of days in each place must not surpass this number. This parameter also allows us to avoid the model to only consider one city (with the best score and lowest living costs) by forcing the change after a certain quantity of days.

Minimum number of days: This number decide how much of a “nomad” can a traveler be, given a minimum threshold of days that can result in constant displacements. IN the opposite way, if it has a high value, the movement between places will be harshly reduced. This can e useful for travelers that don´t mind moving frequently or minimizing their displacements (e.g. Travelers with children, motor liabilities or the elderly).

III. OBJECTIVES STABLISHMENT

1. Identify the best travelling plan given a set of places (cities) and the other proposed restrictions.
2. Identify and implement a metaheuristic that allows the program to find good solutions to the problem given a certain execution time threshold.

IV. MODEL FORMULATION AND PLANNING

A. Model conceptualization

Parameters:

- mind; Minimum number of days.
- maxd: Maximum number of days.
- s: Starting point.
- d: Number of days.
- CT: Transport costs matrix for going from one city to another. Currently only terrestrial means are taken into account.
- CV: Average living costs array for every city.
- n: Number of cities.
- p_1: Priority for maximizing the number of days.
- p_2: Priority for minimizing the Budget spent.
- S: Scores matrix for city i with an interest point l.
- R: Quantity of reviews matrix for city i with an interest point l.

Decision variable:

$$x_{ij} \in \mathbb{B}^{dxn}$$

In day *i*, the traveler stays in city *j*.

Target function: W estate a multi-objective model using the method of pondered summation of weights between the two proposed functions: quantity of days and spent Budget (divided in transport and living costs).

$$\min(p1 * F1 + p_2 * F2)$$

(1)

Where:

$$F1 = \frac{(\min(Puntaje) * d + 1)}{\sum_{i=0}^{d-1} \sum_{j=1}^{n} x_{ij} * Puntaje(j) + 1}$$

F2

$$= \frac{\left[\sum_{i=0}^{d-2} \sum_{j=1}^{n} \sum_{k=1,\ k\neq j}^{n} x_{ij} \cdot x_{i+1\ k} \cdot CT_{jk} + \sum_{i=0}^{d-1} \sum_{j=1}^{n} x_{ij} \cdot CV_j\right]}{(\max(CV) + \max(CT)) * d}$$

$$Puntaje(i) = \sum_{l}^{2*\max d} \frac{S_{il}}{5} * \sqrt{R_{il}} \ \ (2)$$

Formula () assigns a score to a destiny l proportional to his aggregated rating (average) and the number of persons that have rated the destination. The formula “normalizes” scores dividing them by 5 and seeks to penalize those destinations with a huge amount of reviews by calculating the squared root of such values. Due to the fact that there can be places without a huge popularity that should be considered in the travelling plan, this penalty is taken into consideration.

In equation (1), we decided to normalized both functions F1 (quantity of places) and F2 (costs function) with the use of two factors:

A. F1: Multiply by the minimum of the scores calculated in “Puntaje” times the number of days, plus 1

$$(Minimum\ of\ "puntaje" * d) + 1$$

B. F2: Divide by the summation of the maximum cost of transport with the maximum average living cost, times d.

$$(Maximum\ Transport\ Cost + Maximum\ Daily\ Cost) * d$$

Restrictions:

1) The costs structure for going from one place to another must be represented in real numbers. Their value will always be greater or equal than zero.

$$CT \in \mathbb{R}^{nxn},$$
$$\left(\forall \mathrm{i,j} \middle| 1 \le \mathrm{i} \le \mathrm{n} \wedge 1 \le \mathrm{j} \le \mathrm{n}: \mathrm{CT_{ij}} \ge 0\right) (3)$$

2) The structure of costs for average living by day is represented with real numbers with values greater or equal than zero.

$$CV \in \mathbb{R}^{n}, (\forall \mathrm{i} | 1 \le \mathrm{i} \le \mathrm{n}: \mathrm{CV_i} \ge 0) (4)$$

3) The scores structure is represented with real numbers ranging from 1 to 5.

$$S \in \mathbb{R}^{nxl},$$
$$\left(\forall \mathrm{i,j} \middle| 1 \le \mathrm{i} \le \mathrm{n} \wedge 1 \le \mathrm{j} \le 2 * \mathrm{maxd} : 1 \le \mathrm{S_{ij}} \le 5\right) (5)$$

Where l is the predefined number of points of interest per city.
l

4) The structure of quantity of reviews is represented with real numbers bigger than or equal to zero.

$$R \in \mathbb{R}^{nxl}$$
$$\left(\forall \mathrm{i,j} \middle| 1 \le \mathrm{i} \le \mathrm{n} \wedge 1 \le \mathrm{j} \le 2 * \mathrm{maxd} : \mathrm{R_{ij}} \ge 0\right) (6)$$

Where l is the predefined number of points of interest per city.
l

5) The summation of the priorities must be equal to 1.

$$p_1 + p_2 = 1\ (7)$$

6) The priorities are rational numbers between 0 and1:

$$(\forall i | 1 \le i \le 2: p_i \in Q \wedge 0 \le p_i \le 1) (8)$$

7) The person cannot surpass his minimum and maximum threshold of days per city during the journey:

$$\left(\forall j \middle| 1 \le j \le n: \textstyle\sum_{i=1}^{d} x_{ij} * \left(\sum_{i=1}^{d} x_{ij} - mind\right) \ge 0\right) (9)$$
$$\left(\forall j \middle| 1 \le j \le n: \textstyle\sum_{i=1}^{d} x_{ij} \le maxd\right) (10)$$

8) The person cannot go to two cities on the same day:

$$\left(\forall i \middle| 0 \le i \le d-1: \textstyle\sum_{j=1}^{n} x_{ij} = 1\right) (11)$$

9) The first day (i=0) the person must go to someplace:

$$x_{0s} > 0 \quad (12)$$

B. Data Adquisition

Cost of living per city was extracted from Numbeo [2] and transport costs were extracted from Rome2Rio [4]. Specifically, data was retrieved manually but both sources expose API's for the automation of this process, which we left for a further iteration as Rome2Rio API requires their approval.

An important consideration is that transport between any cities a and b is the cheapest found in the Rome2Rio 'query' from city a to city b. As this cost might not be the same on both directions, a to b might have a different cost than b to a.

V. MODEL TRADUCTION

To solve this complex problem, we translated the previous mathematical model into the evolutionary algorithm SPEA 2 [6] (a metaheuristic) in order to achieve quickly almost-optimal solutions. Additionally, the following design decisions were taken into account:

A. Chromosome: <Allele on day 1> - <Allele on day 2>- …. <Allele on day d>

Allele: < Id of the city visited >

B. Euclidian distance with the two target functions:

$$d_{ij} = \sqrt{(f_{1j} - f_{1i})^2 + (f_{2j} - f_{2i})^2}$$

C. Since SPEA 2 already calculates the whole Pareto Front, the use of parameters P1 and P2 are no longer necessary.

D. Truncation operator in environmental selection consists on removing the least-fit items on $\bar{P}_{t+1}$ until $|\bar{P}_{t+1}| = \bar{N}$

E. Variation recombination was applied with double cut crossover (with allele cut points chosen randomly) and uniform crossover. Mutation operator consisted on changing an allele's city to another random city, with a probability of 10%, iterating through every allele of the chromosome.

F. As P, $\bar{P}$ and A are sets, they don't have duplicate chromosomes.

G. Termination was only limited to a maximum number of generations. This is left variable as it heavily impacts execution time.

VI. VERIFICATION AND VALIDATION

For the verification process, we used the language GAMS with the development environment "gamside" that allowed the proper translation of the mathematic model in order to solve the problem with mathematical methods that, despite not being optimal in time, is able to find answers for the basic scenarios of the previously stated model through the use of a mixed integer non-linear problem solver (MINLP) like BONMIN (only used for the base scenario 4) or COUENNE.

In order to validate the metaheuristic mentioned in section V, multiple input files were created for our verification program in GAMS that would represent a series of scenarios with their respective parameters theoretically calculated without the help of any external source. For each one of such scenarios, a test was done with the code in both GAMS (Using the pareto front model in order to compare the results with the metaheuristic) and Java (SPEA2). The results of these tests are:

A. Base Scenario 1 (Basic functioning)

1) P2=0

2) Mind=1 and Maxd=1

3) d=4

4) n=4

5) The rest is random

EXPECTED: Travel to each city in one day.

GAMS:

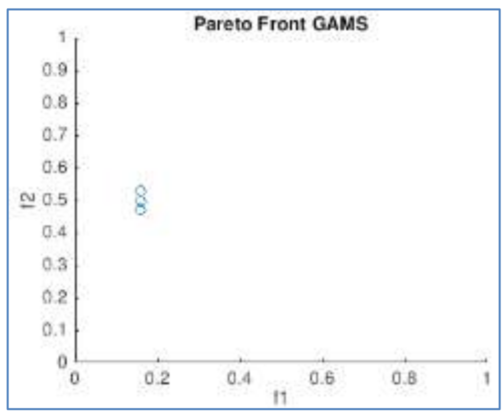

Fig 1. Base scenario 1 results using GAMS

SPEA 2 (Java):

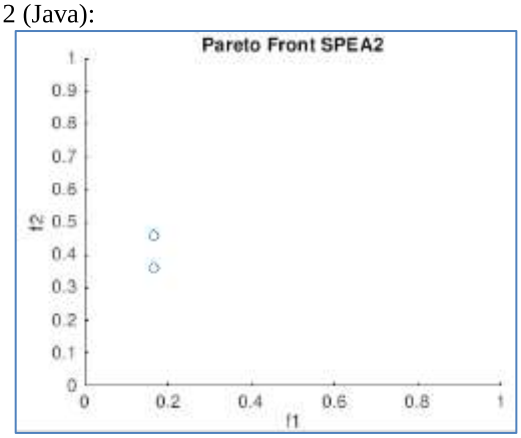

Fig 2. Base scenario 1 results using SPEA2

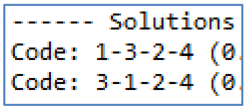
------ Solutions
Code: 1-3-2-4 (0
Code: 3-1-2-4 (0

Fig 3. Base scenario 1 chromosome results using SPEA2

B. Base Scenario 2 (Number of days):

1) n=2
2) mind=3
3) maxd=5
4) d=8
5) s=1
6) Puntaje(1)=10
7) Puntaje(2)=1
8) p1=1
9) p2=0
10) The rest is random

EXPECTED: 5 days in the first city, and 3 in the other one.
GAMS:

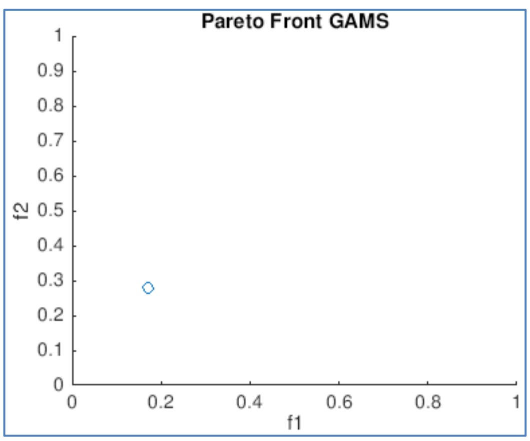

Fig 4. Base scenario 2 results using GAMS

SPEA 2 (Java):

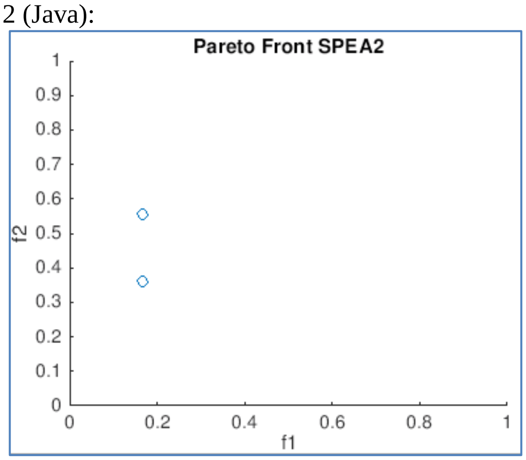

Fig 5. Base scenario 2 results using SPEA2

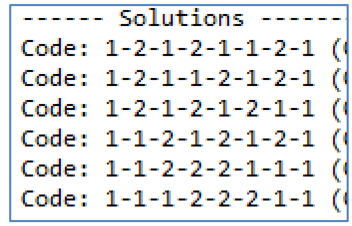
------ Solutions ------
Code: 1-2-1-2-1-1-2-1 (
Code: 1-2-1-1-2-1-2-1 (
Code: 1-2-1-2-1-2-1-1 (
Code: 1-1-2-1-2-1-2-1 (
Code: 1-1-2-2-2-1-1-1 (
Code: 1-1-1-2-2-2-1-1 (

Fig 5. Base scenario 2 chromosome results using SPEA2

C. Base Scenario 3 (Average living cost):

1) n=2
2) mind=3
3) maxd=5
4) d=8
5) s=1
6) $CV_1 = 1$
7) $CV_2 = 10$
8) The values in the transport costs matrix are the same.
9) p1=0
10) p2=1
11) The rest is random.

EXPECTED: 5 days in the first city, 3 in the other one.
GAMS:

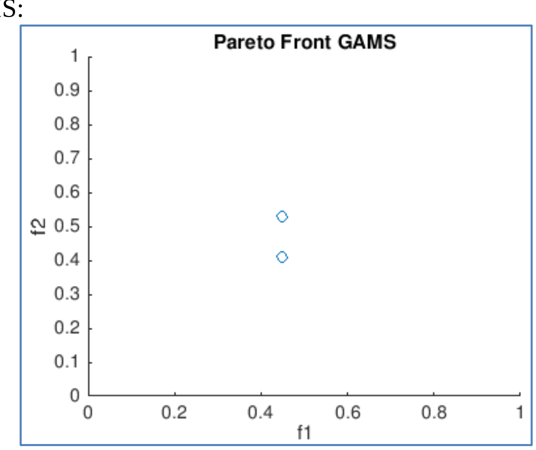

Fig 6. Base scenario 3 results using GAMS

SPEA 2 (Java):

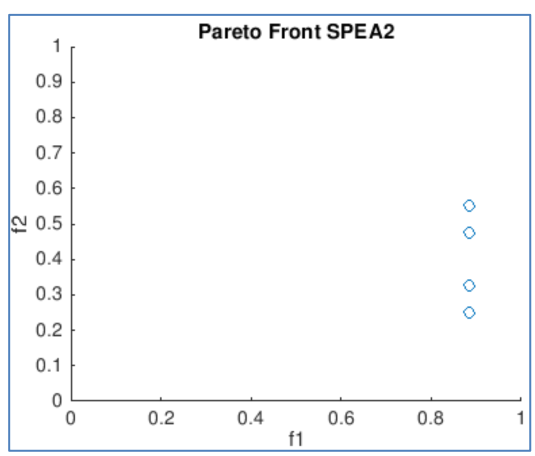

Fig 7. Base scenario 3 results using SPEA2

```
------ Solutions -------
Code: 2-1-2-1-2-1-1-1 (0
Code: 1-2-2-2-1-1-1-1 (0
Code: 1-1-1-2-2-1-2-1 (0
Code: 1-2-2-1-2-1-1-1 (0
Code: 1-2-1-2-2-1-1-1 (0
Code: 2-2-2-1-1-1-1-1 (0
```

Fig 8. Base scenario 3 chromosome results using SPEA2

D. Base Scenario 4 (Transport costs):

1) n=4
2) mind=|
3) maxd=1
4) d=4
5) s=1
6) p2=1
7) p1=0
8) *CT*:

Infinite	10	100000	Infinite
10	Infinite	2	Ifinite
9999	12	Infinite	1
Infinite	Infinite	Infinite	Infinite

Fig 9. Base scenario 4 costs table.

9) Average living cost is the same for every city.
10) The rest is random

EXPECTED: Goes from city 1 to 2, then from 2 to 3 and finally from city 3 to 4.
GAMS:

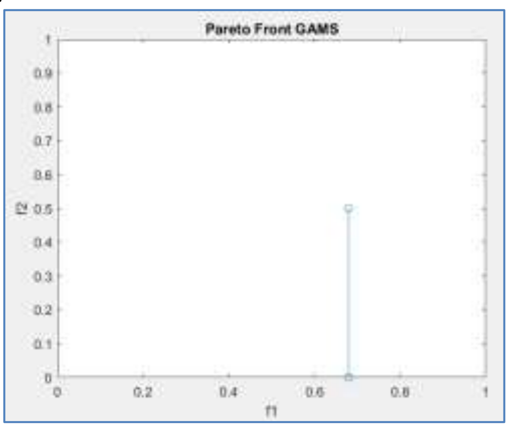

Fig 10. Base scenario 4 results using GAMS

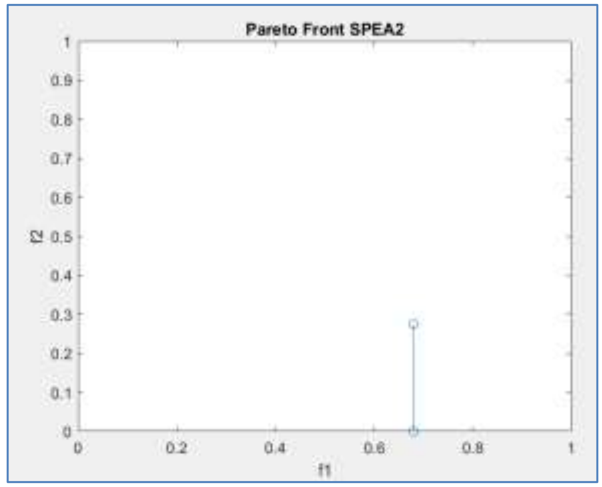

Fig 11. Base scenario 4 results using SPEA2

```
------ Solutions
Code: 1-2-3-4 (0.
Code: 1-3-4-2 (0.
```

Fig 12. Base scenario 4 chromosome results using SPEA2

E. Intermediate Scenario:

1) n=10
2) 5 cities have high living cost and low scores (From city 1 to 5)
3) 5 cities have low living cost and high scores (From city 6 to 10).
4) Transport costs are the same.
5) d=15
6) maxd=3
7) mind=1
8) p1=0.5
9) p2=0.5
10) s: One of the cheap cities (from 6 to 10).
11) The rest is random

EXPECTED: Go to the cheapest cities.
GAMS:

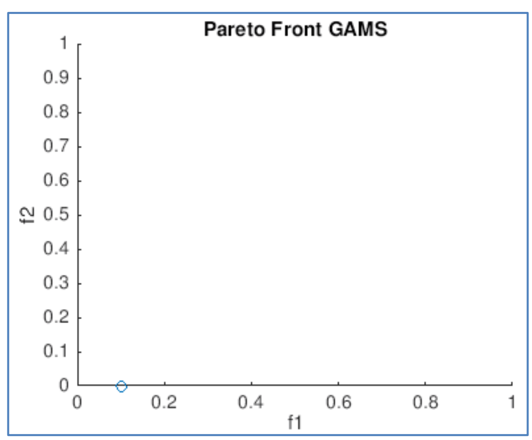

Fig 13. Intermediate scenario results using GAMS

SPEA 2 (Java):

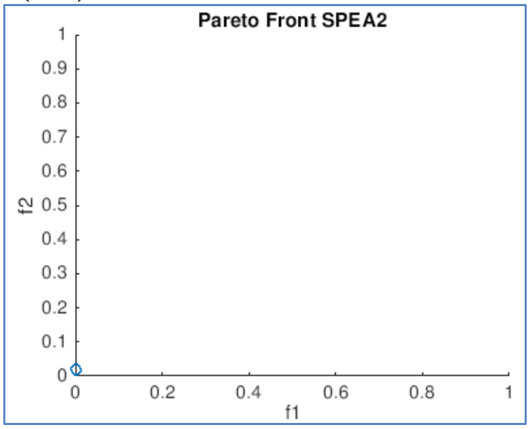

Fig 14. Intermediate scenario results using SPEA2

```
------ Solutions -------
Code: 7-9-8-10-7-6-9-9-10-8-10-6-8-7-6
Code: 8-8-8-10-7-9-9-9-10-10-6-7-7-6-6
Code: 8-9-8-10-7-6-9-9-10-10-6-7-8-7-6
Code: 8-9-8-10-7-6-9-9-10-8-10-6-7-7-6
Code: 8-8-8-10-7-9-9-9-10-10-6-6-7-7-6
Code: 8-8-8-10-6-9-9-9-10-10-6-7-7-7-6
```

Fig 15. Intermediate scenario chromosome results using SPEA2

F. Real life Scenario:

1) n=16. The researched cities from Spain are in order:

i. valencia
ii. barcelona
iii. sevilla
iv. granada
v. málaga
vi. córdoba
vii. cádiz
viii. almería
ix. toledo
x. salamanca
xi. madrid
xii. zaragoza
xiii. pamplona
xiv. bilbao
xv. segovia
xvi. santander

2) Transport costs are researched in “Rome2Rio”.
3) Average living costs are researched in “Numbeo”.
4) Scores and quantity of reviews are researched in Google.
5) d=30
6) maxd=5
7) mind=2
8) The rest is random

EXPECTED: Unknown

GAMS: Unknown

SPEA 2 (Java):

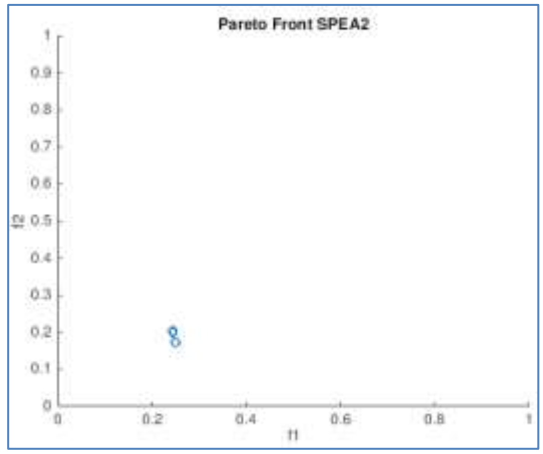

Fig 16. Real life scenario results using SPEA2

```
- Solutions -------
1-14-2-6-12-2-3-4-12-14-6-2-11-11-2-12-3-3-2-3-4-1-11-3-1-11-1-14-11-1 (
1-14-2-1-12-2-3-12-12-14-9-2-11-11-2-9-3-3-2-3-4-1-11-3-4-11-14-4-11-1 (
1-14-2-1-12-2-3-12-12-14-9-2-11-11-2-9-3-3-2-3-4-12-11-3-4-11-1-14-11-1
1-14-2-1-12-2-3-12-12-14-9-2-11-11-2-9-3-3-2-3-4-12-11-3-1-11-14-4-11-1
1-14-2-1-12-2-3-12-12-14-9-2-11-11-2-9-3-3-2-3-4-1-11-3-1-11-14-4-11-1 (
1-14-2-1-12-2-3-12-12-14-9-2-11-11-2-9-3-3-2-3-4-1-11-3-4-11-1-14-11-1 (
```

Fig 17. Real life scenario chromosome results using SPEA2

VII. RESULTS ANALYSIS

In most cases the SPEA2 algorithm presents a points that belong to the pareto front constructed in GAMS. However, there are 2 exceptions in the scenarios stated in this document:

A. Base Scenario 3: Even though the values of f1 vary in a multiplying factor of almost 2, the answer presented by SPEA 2 is valid for the given scenario, since it sends the person to the first city five times in all the given responses. It is believed that the solutions given by the metaheuristic are not the best in terms of costs, but they fulfill the conditions of the problem.

B. Intermediate Scenario: Although the SPEA 2 algorithm does not given in this case the “optimal” solutions, it solves the given problem with the expected result: to send the person through the cheapest and top scored cities. For this reason, the answer given is valid.

For the rest of the cases, the evolutionary algorithm succeeds in its implementation by giving another alternative to the resolution of the mathematical model through the use of numerical methods (GAMS). In non-trivial cases, like the intermediate and real life scenarios, it has a faster processing time compared to the model developed with “gamside” (For instance, with “gamside” we could not run our real life scenario, since after 45 minutes it got stuck in a loop and assigned a value of zero for all the values in our decision variable X).

VIII. CONCLUSIONS AND FUTURE WORK

A. The SPEA2 implementation does not correctly give the pareto front for every situation. Although it presents acceptable solutions, further studies must be known in order to correctly know if its functioning perfectly for every given situation.

B. The metaheuristic effectively reduces the execution time compared to the numerical methods solution with GAMS. Due to this, with some minor adjustments, this model could become a commercial product for using in the industry.

C. For future work, this model could include the use of itineraries for activities inside the given cities, to have a full-time lanner that does not only tell you where to stay but also what to do in each day.

D. For another future work, another could be implemented in which, given a maximum budget, it calculates the longest journey it can make with such restriction.

BIBLIOGRAPHY

[1] W. Coldwell, “10 of the best travel apps ... that you'll actually use: part two,” The Guardian, 06-Feb-2017. [Online]. Available: https://www.theguardian.com/travel/2017/feb/06/top-10-useful-travel-apps. [Accessed: 09-Dec-2018].

[2] Numbeo, “Cost of Living,” Numbeo. [Online]. Available: https://www.numbeo.com/cost-of-living/. [Accessed: 09-Dec-2018].

[3] M. Patkar, “5 Smart Travel Planning Apps for Easier Trip Itineraries,” MakeUseOf, 17-Mar-2018. [Online]. Available: https://www.makeuseof.com/tag/smart-travel-planning-apps/. [Accessed: 09-Dec-2018].

[4] Rome2Rio, “De Sevilla a Barcelona en 3½ horas: precios y horarios para ir en Tren, Autobús ó Coche compartido,” Rome2rio. [Online]. Available: https://www.rome2rio.com/es/map/Sevilla/Barcelona. [Accessed: 09-Dec-2018].

[5] M. Rouse, “What is Google Maps? - Definition from WhatIs.com,” WhatIs.com, Feb-2013. [Online]. Available: https://whatis.techtarget.com/definition/Google-Maps. [Accessed: 09-Dec-2018].

[6] E. Zizzler, M. Laumanns, and L. Thiele, “SPEA2: Improving the Strength Pareto Evolutionary Algoithm,” May-2001. [Online]. Available: https://pdfs.semanticscholar.org/6672/8d01f9ebd0446ab346a855a44d2b138fd82d.pdf. [Accessed: 09-Dec-2018].