Tourist Trip Route Problem

Alejandro Marrero Díaz
email alu0100825008@ull.edu.es

La Laguna, Tenerife, ES

Abstract

In this document I am going to outline the workflow, solutions and results to the Tourist Trip Route Problem (TTRP) [12, 1] proposed as a assignmment of the subject Intelligent Systems of the Master of Computing Science of the University of La Laguna. Basically, the TTRP is an extension of the common problem Traveling Salesman Problem (TSP) where each point or location has a rating and the goal is to create the better route possible according to that. For this work, some metaheuristics as Local Search (LS) [9], Simulated Annealing (SA)[3, 10, 6, 4], Variable Neighbordhood Search (VNS) [5] and Tabu Search (TS)[2] has been implemented to solve the TTRP [12, 1].

Keywords: Computing Science, Tourist Trip Route Problem, Metaheuristics

1. Data Collection Process

The process of collecting data was pretty rudimentary. Simply, I just select thirty different places in Tenerife from the web TripAdvisor and, after that I just calculated the distance between each place with the Google Maps Tool. Due to the fact that this work is only for educational purpose, I assume that the distance between to places A and B where the same, so, the distance from A to B is the same that the distance from B to A. And, with this, I am able to store the distances into a unidimensional array doing the compute process easier. Then, here it is shown the list of locations used in this work:

ID	Name	Category	Rating	Duration (minutes)
0	Hotel Riu Garoe	0	5	0
1	Iberostar Grand Hotel Salome en Costa Adeje	0	5	0
2	Finca Buenavista	0	5	0
3	Loro Parque	2	4.5	360
4	Siam Park	2	4.5	360
5	Avistamiento delfines en Las Américas	5	3	90
6	Playa de Benijo	1	5	180
7	Playa de Las Teresitas	1	3	180
8	Playa de San Telmo	1	2	180
9	Kite Surf en Adeje	5	5	120
10	Seascuba en Santiago del Teide	5	5	60
11	Kayaking en Adeje	5	4	90
12	Motos de agua en Las Galletas	5	4	45
13	Paseo en Helicoptero en Adeje	6	4.5	30
14	Teide	3	5	120
15	Barranco de Masca	4	4.5	210
16	Kayaking en Los Gigantes	5	3.5	120
17	Rambla de Castro en Los Realejos	1	2.5	240
18	Reserva de San Blas en San Miguel de Abona	7	4	120
19	Las Águilas-Jungle Park en Arona	2	4	180
20	Paseo en Camello en Arona	6	3.5	60
21	Basílica de Nuestra Señora de La Candelaria	6	4.5	30
22	Jardín Botánico	7	3	30
23	Jardín Victoria	7	4.5	30
24	Lago Martianez	5	4.5	180
25	Punta de Teno	3	4.5	180
26	Montañas de Anaga	4	4.5	300
27	Observatorio del Teide	8	4	120
28	Museo de la Naturaleza y el Hombre	8	4	60
29	Casa de los Balcones	8	4	30

Table 1: Locations

And here there is the table of categories reference:

ID	Category
0	Hotel
1	Playa
2	Zoo
3	Mirador
4	Senderismo
5	Acuatica
6	Entretenimiento
7	Paseo
8	Museo

Table 2: Categories

2. Trourist Trip Route Problem Design

As I have said before, due to the fact that this work has only educational purpose, I have considered an lightweight version of the Tourist Trip Route Problem [12, 1]. Hence, in this implementation I have only considered the following variables:

- Number of days.
- Amoung of hours avaibles for driving each day.
- Circular or lineal routes.
- Point of start.

So, in an effort of getting better results of the performance of the developed algorithms, I decided to set up the following configuration:

- Number of days = 5 days.
- Hours driving per day = 10-12 hours.
- Circular routes for tourists who lodge in a hotel.
- Point of start = Hotel where the tourist is roomed.

3. Algorithms

I this section, I am not going to explain each algorithm I have implemented for this work, but I am going to explain what modifications or neighbordhood structures I have used seeking to increase the performance of these algorithms.

3.1. Initial Solution Generation

All the implemented algorithms must start from any initial feasible solution, so, for this work I have considered two ways of create that initial solution.
On the one hand, I have developed a Greedy algorithm which creates an initial solution considering only the stars of every place.
On the other hand, a random solution was created considering a weighing equation an applying the Opposite-Based Learning technique.

3.1.1. Opposition-Based Learning

Opposition-based Learning (OBL) [7, 11, 8] is a computing concept which has demostrated great efectivity at the time of improve several optimization algorithms. When we are evaluating a solution X, which belongs to the set of feasibles solutions S, simultaneously, we calculate the opposite solution $\overline{X}$, in order to achieve a better exploration of the search space Ω looking for the global optima [11].

Being $x \in \Re$ a real number defined within a certain range $x \in [a, b]$. The opposite number of X, denoted as $\overline{x}$ is defined as follows [11, 7]:

$$\overline{x} = a + b - x \tag{1}$$

Taking into acount that, in this problem we are not working with real numbers but places, the way we calculate the opposite place of any X place is the same but, in this case, the range $[a, b]$ is the number of possible places we can select in the problem.

Finally, the place inserted in the initial solution is the one which has better rate when aplying the following equation:

$$\frac{0.3 * X_{duration} + 0.7 * X_{stars}}{X_{stars}^2} \tag{2}$$

3.2. Neighbordhood Structures Used

The next stage after calculating the initial solution, is trying to improve it using the neighborhood structures, looking for neighbord solutions which have better evaluation. Eventhough there are many possible neighbordhood structures to test, I decided to use the following ones:

- **Swap**: It tries to swap a random solution-included point for any random non-included point without exceed the route duration limit.
- **Insert**: It inserts a random non-included solution point in the new solution if it does no exceed the route duration limit.
- **Remove**: It simply removes any random point from the solution, without being the inital point.

After doing any of these changes, I calculate the evaluation of the new solution and evaluate which one, the new one or the old one, must become the actual solution for the next iteration considering they evaluations.

4. Results

For this study, I chose the following *Tourist Trip Route Problem* configuration:

- 5 days of trip.
- Circular route from the start point. In this case the start point of every route is the Hotel Riu Garoe in Puerto de La Cruz.
- No more than ten hours per route. This was setting like that in order to have more possible solutions for every route.

According to this configuration, I have test all the implemented algorithms with the following configurations:

- **Local Search (LS)**: the only parameter of this algorithm is the number of chances I give it to choose a worse solution trying to scape for any local optima. In this case, the number of chances were 2.
- **Variable Neighbordhood Search (VNS)**: This algorithm only have the k parameter, which sets the number of neighbordhood structures. In this case, k was equal to 3.
- **Simulated Annealing (SA)**: initial temperatura equal to 50 and a decrease rate of 0.9 in each iteration.
- **Tabu Search (TS)**: For the Tabu Search algorithm I have only set the number of checked solutions which shape the tabu list. In this case, the size of the tabu list was only 4 solutions due to the low number of possible points.

Finally, every above configuration was testing using the *Greedy Inital Solution* and the *Opposition-Based Learning (OBL) Initial Solution* seeking obtain the better performance possible. Below are the tables with the obtained results, where it is the mean and standard deviation of every experiment considering the duration in minutes and the rating of the calculated routes.

	OBL Initial Solution								Greedy Initial Solution							
	Local Search		*VNS*		*SA*		*Tabu Search*		*Local Search*		*VNS*		*SA*		*Tabu Search*	
	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ
	503	77,3369252	**502.6**↑	**55,43735203**↑	523	79,11384203	562.2	33,72980878	582	14,24780685	**472,4** ↑	**157,9898731**↑	528.8	112,9522023	582	14,24780685

Table 3: Performance results considering routes duration in minutes

	OBL Initial Solution								Greedy Initial Solution							
	Local Search		*VNS*		*SA*		*Tabu Search*		*Local Search*		*VNS*		*SA*		*Tabu Search*	
	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ
	18	7,474958194	17,6	4,083503398	17	5,744562647	**18.8** ↑	**3,962322551**↑	18,4	7,4615682	**18,7**↑	**7,233602146**↑	18,4	7,4615682	18,4	7,4615682

Table 4: Performance results considering routes rating

As we can see, the *Variable Neighbordhood Search* algorithm gets the best results for duration variable with the *Greedy Inital Solution* technique but the *Tabu Search* algorithm gets the best solutions for the rating variable using the *OBL Initial Solution* approach.

Hence, considering the given results and a ratio $\frac{rating}{duration}$, after doing the maths I conclude that the algorithm **Variable Neighbordhood Search (VNS)** using the **Greedy Initial Solution** generation strategy shows the better performance. Here it is shown the results:

Greedy Initial Solution:	**OBL Initial Solution:**
Local Search = 0,03161512	Local Search = 0,035785288
VNS = **0,039585097** $\uparrow$	VNS = **0,035017907**
SA = 0,034795764	SA = 0,03250478
Tabu Search = 0,03161512	Tabu Search = 0,033440057

5. Encountered Problems

Throughout this work I had only have code issues when I was trying to implement an accurate software design to the proposed problem. Finally, I solved it creating two classes, one of them for the distance matrix and the other one for the locations information.
Afterwise, an abstract metaheuristic class was developed in order to avoid repeated code in every algorithm implemented. So, every metaheuristic algorithm inherit from the metaheuristic mother class and implemented it's owm features

6. Bibliography

[1] J. Brito, A. Expósito-Márquez, and J.A. Moreno. A fuzzy grasp algorithm for solving a tourist trip design problem. 2017.

[2] Ke-Lin Du and M. N. S. Swamy. *Search and Optimization by Metaheuristics*. 2016.

[3] M. Gerber and L. Bornn. Improving simulated annealing through derandomization. *Journal of Global Optimization*, 68(1):189–217, 2017.

[4] M. Gerber and L. Bornn. Improving simulated annealing through derandomization. *Journal of Global Optimization*, 68(1):189–217, 2017.

[5] Pierre Hansen, Nenad Mladenović, and José A. Moreno Pérez. Variable neighbourhood search: methods and applications. *Annals of Operations Research*, 175(1):367–407, 2010.

[6] G. Karagiannis, B.A. Konomi, G. Lin, and F. Liang. Parallel and interacting stochastic approximation annealing algorithms for global optimisation. *Statistics and Computing*, 27(4):927–945, 2017.

[7] Alejandro Marrero Díaz. Desarrollo de algoritmos dirigidos por retos. Master's thesis, Universidad de La Laguna, July 2017.

[8] Shahryar Rahnamayan, Hamid R. Tizhoosh, and Magdy M. Salama. Opposition-based differential evolution. *Studies in Computational Intelligence*, 143(1):155–171, 2008.

[9] Helena Ramalhinho-Lourenço, Olivier C. Martin, and Thomas Stützle. Iterated local search. Economics Working Papers 513, Department of Economics and Business, Universitat Pompeu Fabra, November 2000.

[10] M.C. Robini, M. Ozon, C. Frindel, F. Yang, and Y. Zhu. Global diffusion tractography by simulated annealing. *IEEE Transactions on Biomedical Engineering*, 64(3):649–660, 2017.

[11] H R Tizhoosh. Opposition-Based Learning: A New Scheme for Machine Intelligence. *Computational Intelligence for Modelling, Control and Automation, 2005 and International Conference on Intelligent Agents, Web Technologies and Internet Commerce, International Conference on*, 1:695–701, 2005.

[12] W. Zheng, Z. Liao, and J. Qin. Using a four-step heuristic algorithm to design personalized day tour route within a tourist attraction. *Tourism Management*, 62:335–349, 2017.