Robust Bi-objective Shortest Path Problem in Real Road Networks

Christian Cintrano(✉), Francisco Chicano, and Enrique Alba

Departamento de Lenguajes y Ciencias de la Computación,
Universidad de Málaga, Andalucía Tech., Málaga, Spain
{cintrano,chicano,eat}@lcc.uma.es

Abstract. Road journeys are one of our most frequent daily tasks. Despite we need them, these trips have some associated costs: time, money, pollution, etc. One of the usual ways of modeling the road network is as a graph. The shortest path problem consists in finding the path in a graph that minimizes a certain cost function. However, in real world applications, more than one objective must be optimized simultaneously (e.g. time and pollution) and the data used in the optimization is not precise: it contains errors. In this paper we propose a new mathematical model for the robust bi-objective shortest path problem. In addition, some empirical studies are included to illustrate the utility of our formulation.

Keywords: Robustness · Traffic road network · Bi-objective shortest path · Multi-objective optimization

1 Introduction

Road trips are an inherent part of our modern lifestyle. To give a quick example, only in U.S.A., citizens spend almost two hours a day driving. We must add to this time the economic (fuel, vehicle maintenance, ...) and psychological (stress, anxiety, ...) costs. For this reason, it is not surprising that the problem of finding the shortest or fastest paths between two points in the city is a popular and well studied problem in the scientific literature.

A rather generalized way of modeling the city's road network is to treat it as a weighted directed graph. The streets are the arcs of this graph, the city intersections are the nodes, and the weights could be the length of the road, speed limits, etc. If we use this model, well-known algorithms like Dijkstra [1] or A* [2] are useful even for commercial applications. However, these models have a very simplistic view of the city, offering as a solution a single route that (generally) only minimizes time (or distance).

But time is not the only objective for the citizens. Fuel, modern environmental concerns, or reduced traffic jams are also new goals to be taken into account for smart mobility in cities. A good point to start with new formulations is to consider the bi-objective shortest path (BSP) problem, that has received special attention in recent years both in academy and industry [3].

E. Alba et al. (Eds.): Smart-CT 2017, LNCS 10268, pp. 128–136, 2017.
DOI: 10.1007/978-3-319-59513-9_13

To make a difference, it is important that the proposed solutions for the BSP problem are applicable in the real world. This requires taking into account real-time and precise information on the state of the roads, traffic jams, flows of vehicles, etc. The challenge of the use of these data sources falls in its dynamic nature, e.g., traffic in rush hour differs from that in the evening, or in its stochasticity, e.g., an accident happens on the road. These are the reasons why a useful solution (algorithm) must take these imprecisions into account and avoid providing routes that, for example, can cause by themselves additional delays in the arrival to the destination. The ability of a solver to tolerate these inaccuracies is called *robustness*. And a variant of the BSP problem that takes into account inaccuracies in its input data will be called *robust bi-objective shortest path problem* (RBSP problem).

In this article we will present a new model of RBSP problem that deals with the robustness from a stochastic point of view. We assume that the input data behaves as probability distributions. Also, we decided work with the bi-objective version of SP problem given that most of the existing instances for this kind of problem optimize only one or two objectives. Therefore BSP problem is the natural choice on which to make our robust model.

The article is structured as follows: Sect. 2 shows the basic formulation of the RBSP problem. Section 3 describes our proposal for modeling the robustness of the problem. Section 4 presents some empirical results of our selected model. Section 5 presents alternative formulations of the robustness for the RBSP problem existent in the state-of-the-art. Finally, Sect. 6 presents the main conclusions of this article and the lines of future work.

2 Basic Formulation

Before presenting our model, it is necessary to give some basic notions about the BSP problem and about the robustness in optimization problems. Then, we formulate the mono-objective and bi-objective shortest path problem. Later, general considerations about robustness in optimization problems are presented.

2.1 Mono- and Bi-objective Shortest Path Problems

Let $G(N, A)$ be a directed graph, where N is the set of nodes, and A the set of arcs between nodes, $A \subseteq N \times N$; we define a path $p = p_1, p_2, \ldots, p_k$ as a list of nodes that: $\forall p_i, p_{i+1} \in p, 0 \le i \le k-1, (p_i, p_{i+1}) \in A$. We define $\mathcal{P}_{s,e}$ as the set of all possible paths between a start node s and an end node e.

We can define a cost function $C : A \to \mathbb{R}^+$ in the graph G. This function assigns a non-negative numeric weight value to each edge of the graph. In order to simplify the formulation, we write $C((i, j)) = c_{ij}$ as the cost of the arc (i, j).

The path between two given nodes s and e in the graph that minimizes the total cost of the path is the result of solving the following problem:

$$\min_{p \in \mathcal{P}_{s,e}} \boldsymbol{z}(p) = \sum_{(i,j) \in p} c_{ij} \qquad (1)$$

where $\boldsymbol{z}(p)$ is the objective function, s and e are (respectively) the start and end nodes of the path. Abusing of notation we write $(i,j) \in p$ when the arc (i,j) appears in the path p.

We will consider, without loss of generality, that the objective (components of the vector function) are to be minimized. Next, we include the definition of some standard multi-objective concepts to make the paper self-contained.

Definition 1 (Dominance). *Given a vector function* $\mathbf{f} : \mathcal{P}_{s,e} \to \mathbb{R}^d$*, we say that solution* $x \in \mathcal{P}_{s,e}$ dominates *solution* $y \in \mathcal{P}_{s,e}$*, denoted with* $x \prec_{\mathbf{f}} y$*, if and only if* $f_i(x) \leq f_i(y)$ *for all* $1 \leq i \leq d$ *and there exists* $j \in \{1, 2, \ldots, d\}$ *such that* $f_j(x) < f_j(y)$*. When the vector function is clear from the context, we will use* $\prec$ *instead of* $\prec_{\mathbf{f}}$*.*

Definition 2 (Pareto Optimal Set and Pareto Front). *Given a vector function* $\mathbf{f} : \mathcal{P}_{s,e} \to \mathbb{R}^d$*, the* Pareto Optimal Set *is the set of solutions* P *that are not dominated by any other solution in* $\mathcal{P}_{s,e}$*. That is:*

$$P = \{x \in \mathcal{P}_{s,e} | \nexists y \in \mathcal{P}_{s,e}, y \prec x\} . \tag{2}$$

The Pareto Front *is the image by* $\mathbf{f}$ *of the Pareto Optimal Set:* $PF = \mathbf{f}(P)$*.*

Definition 3 (Set of Non-dominated Solutions). *Given a vector function* $\mathbf{f} : \mathcal{P}_{s,e} \to \mathbb{R}^d$*, we say that a set* $X \subseteq \mathcal{P}_{s,e}$ *is a* set of non-dominated solutions *when there is no pair of solutions* $x, y \in X$ *where* $y \prec x$*, that is,* $\forall x \in X, \nexists y \in X, y \prec x$*.*

Now we can define the bi-objective shortest path (BSP) problem. This problem was defined by Hansen [4] in 1980. Here is a simple definition of the BSP problem:

Definition 4 (Bi-objective Shortest Path). *Let,* $C = (C^1, C^2)$ *a pair of cost functions, and* $c_{ij,k}$ *the cost of the arc* (i,j) *in objective* k*. The bi-objective shortest path problem are defined as follows:*

$$\min_{p \in \mathcal{P}_{s,e}} \boldsymbol{z}(p) = \sum_{(i,j) \in p} (c_{ij,1}, c_{ij,2}) \tag{3}$$

2.2 Robustness

In real problems, one of the biggest problems is the many inaccuracies in the data. Changes in the environment, noisy sensors and actuators, and lack of knowledge of the problem are some examples of the issues that must be faced by algorithms and applications that want to be useful in the real world. These events led to a new optimization paradigm called robust optimization [5]. The lack of precision can be modeled in two different ways, as explained in [6]: (i) stochastic, which models inaccuracies assuming a random variable for each variable; (ii) robust, assume that the value of each variable is obtained from an uncertain set of values.

In this paper we will use probability distributions instead of fixed values in the cost function of the graph. In this work, we call *robustness* to the ability of dealing with inaccuracies.

3 Robustness Model

There are multiple ways to apply robustness to the BSP problem. However, they all have a number of limitations, either by simplifying the real world, or by generating complex objective functions. We here propose a simple model of robustness for this problem based on treating the costs as probability distributions. With this approach, we want to generate a simple objective function that is able to deal with not exact data input.

In real world, much data is uncertain. In the routing of vehicles, the speed of vehicles, the time, the amount of gases emitted, etc. depend on multiple factors. Some examples of such factors are: the hour of the day, the degree of congestion of the road/lanes, the model of vehicle used, or the driver's profile in terms of acceleration up/down. For this reason, for the same trip, these environment variables can take multiple values.

Despite this apparent randomness, environment variables often have a temporary and stationary behavior. Rush hour traffic is usually similar in the same period of the year. Given this behavior we could interpret, for example, travel time as a probability distribution with μ, the average travel time at any hour of the day, and σ^2, its variance. We assume that random variables are independent.

For our RBSP problem, we will transform our fixed arc costs $c_{ij,k}$ into random variables $\hat{c}_{ij,k}$. This transformation of the kth-cost function into the new function will depend on the knowledge of the particular real world variable. In a similar way, we change our objective function $\boldsymbol{z}(p)$ in a new $\hat{\boldsymbol{z}}(p)$

$$\hat{\boldsymbol{z}}(p) = \sum_{(i,j)\in p} \hat{c}_{ij,k} \tag{4}$$

A random variable can have many values in theory. However, with regards to road travel, many factors tend to have similar values. A typical way to be more flexible in the data is consider, e.g., travel time, within a range. But, unforeseen events can always occur that cause the data come out of that range. One way to take all this information into account is to use probability distributions that most closely resemble real behavior. We will characterize each probability distribution by two functions: $\mu(\hat{c}_{ij,k})$ that return the average of the probability distribution, and $\sigma^2(\hat{c}_{ij,k})$ that returns the variance as output. Abusing of notation we write $\mu(\hat{c}_{ij,k})$ as $\mu_{ij,k}$, that means the average of the random distribution of the arc (i, j) in the objective k. In a similar way we write $\sigma^2(\hat{c}_{ij,k})$ as $\sigma^2_{ij,k}$.

For an ideal road trip, its associated cost must be minimal and have the least variation possible. Based on this idea we use, in Eq. 5, our robust objective function to define the RBSP problem.

$$\min_{p\in\mathcal{P}_{s,e}} \boldsymbol{z}^R(p) = \sum_{(i,j)\in p} (\mu_{ij,1}, \mu_{ijm,2}, \sigma^2_{ij,1}, \sigma^2_{ij,2}) \tag{5}$$

The values of μ and σ^2 are the new costs associated to each arc of the graph. In this way, we transform our bi-objective problem into another one with four objectives, being able to apply any general algorithm that solve problems with four objectives.

4 Experimental Study

In this section we will present some experiments to illustrate the use of our model. We will use the real map of the city of Málaga (Spain) to show how our approach is applicable to the real world.

4.1 Algorithms

We test the impact of applying robustness to our problem by using two types of algorithms. On the one hand, we use the mono-objective algorithms Dijkstra and A*, because they are typically used in the state-of-the-art and applications. On the other hand, we use an algorithm to obtain the Pareto optimal set in our four-objective robust model. The selected the state-of-the-art multi-objective algorithm is PULSE [7]. This is an exact recursive algorithm based in aggressive pruning strategies. We implemented PULSE in the Java programming language and included the modifications that the authors proposed in the article to extend it to four objectives.

4.2 Málaga City (Spain)

The city of Málaga is one of the Spanish cities pioneering in smart cities initiatives. It is an example of medium-size European city. Figure 1 shows a map of Málaga. The graph used in this study to model the city was obtained from Open Street Map[1]. This graph had a preprocessing step to became strongly connected and the resulting graph has 45,410 nodes and 118,388 arcs. Thanks to this, our results would be compatible with applications that make use of real maps of Open Street Maps. In this work, since we only want to validate the proposed model, the weights of the arcs were randomly generated.

Fig. 1. Map view Málaga (Spain) from Open Street Map

[1] Official web site: https://www.openstreetmap.org.

4.3 Methodology

In this section we describe the methodology follows in our empirical study. To illustrate the behavior of our model we are going to use algorithms that return solutions without regard to robustness (Dijkstra and A*) and others whose outputs take into account the robustness (PULSE). Because, Dijkstra and A* are mono-objective algorithms, we use weighted sums to take into account two objectives. In both cases, we only assume fixed data in these two mono-objective algorithms, so we use the weighing: $\alpha_1^R + (1 - \alpha)z_2^R$ with the weight α getting the different values: $\alpha = 1$, $\alpha = 0$, $\alpha = 0.5$. With this weighting we want to get a variety of solutions.

We run the algorithms which inputs are a graph (Málaga) and a start and end nodes. In the graph of Málaga we selected a total of five pairs of random nodes. Each node is numbered between 0 and 9. For each pair (i, j) we calculate two paths: one from i to j, and another from j to i.

These paths generate a total of ten instances on which we run the algorithms. We will measure the execution times of each algorithm, because it is desirable that the algorithm be fast, and the four objectives ($\boldsymbol{z}^R$) of each solution obtained: the Pareto set in the PULSE algorithm and the single solution in the mono-objective algorithms. Get $\boldsymbol{z}^R$ will allow us to compare if one solution is more robust than another. In this paper, we will assume that one solution (path) p^1 is more robust than another p^2 if $z_3^R(p^1) < z_3^R(p^2), z_4^R(p^1) < z_4^R(p^2)$.

4.4 Empirical Results

Since the representation of four objective variables is a complex problem, we have opted for a representation in the form of value ranges as it is showed in the Fig. 2. In this, an example of the Pareto front of one of our test paths (3,2) is displayed. The boxes are centered on z_1^R (x-axis) and z_2^R (y-axis) values; and its width of each box expresses the z_3^R (the y-axis is analogous to the x-axis). We can see that Dijkstra is not within the front obtained by PULSE. A*, on the other hand, is on the front, but it is only one of the 21 solutions in the front.

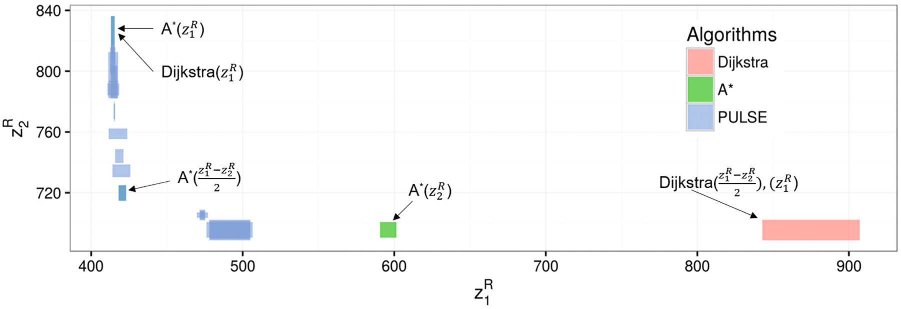

Fig. 2. Pareto front for one path of the Málaga City

Table 1. Computational time in seconds of each algorithm

Path	Dijkstra			A*			PULSE
	z_1^R	z_2^R	$0.5z_1^R + 0.5z_2^R$	z_1^R	z_2^R	$0.5z_1^R + 0.5z_2^R$	
0,1	0.53	4.45	0.14	0.50	4.02	0.29	45.65
1,0	0.05	2.66	0.06	0.07	5.72	0.10	82.89
2,3	1.23	4.29	0.72	1.48	4.09	1.81	4879.25
3,2	0.31	1.26	0.48	0.74	3.39	0.98	3641.96
4,5	1.05	2.87	1.11	0.92	1.58	1.15	1696.29
5,4	0.96	1.01	0.75	1.04	2.63	1.72	11950.93
6,7	1.14	1.26	1.73	1.25	0.81	1.51	1192.76
7,6	0.24	0.14	0.13	0.17	0.33	0.26	1616.03
8,9	0.49	4.12	0.43	0.26	9.87	0.88	115.98
9,8	0.04	0.64	0.04	0.06	2.22	0.28	171.19

Next we will analyze some parameters measured in the different paths. Table 1 shows the execution times of each algorithm to compute each path. As we would expect, PULSE is slower than the other two algorithms. The differences between execution times in the PULSE algorithm are because the proximity between the start and end points. However, while the mono-objective algorithms only found one solution, PULSE finds the whole Pareto set of solutions.

In order to compare the robustness between the obtained solutions, we calculated the number of solutions in the Pareto set whose objectives z_3^R and z_4^R (variances of the two probability distributions) are simultaneously strictly smaller than the solutions obtained by the three versions of Dijkstra and A*. In the Table 2 we observe that almost three solutions with less variability are obtained.

Table 2. Pareto size and number of solution obtained by PULSE more robust than mono-objective algorithms

Path	Pareto set size	Dijkstra			A*		
		z_1^R	z_2^R	$0.5z_1^R + 0.5z_2^R$	z_1^R	z_2^R	$0.5z_1^R + 0.5z_2^R$
0,1	4	0	1	0	0	0	0
1,0	4	0	1	0	0	0	0
2,3	26	1	1	0	1	1	0
3,2	21	3	0	0	3	2	0
4,5	5	0	0	0	0	0	0
5,4	6	1	1	0	1	1	0
6,7	4	0	0	0	0	0	0
7,6	3	0	0	0	0	0	0
8,9	6	1	0	0	1	0	0
9,8	24	8	0	0	8	0	0

While the computing time is higher, there are situations where it is preferable to obtain more predictable routes, i.e., travel to airport or job interviews, salesman that should visit various commercial establishment, etc.

5 Related Work

In tis section we present some works related to the application of robustness to SP and BSP problem. First, the problem of the mono-objective SPP with robustness is well studied in the literature. There are many models that add robustness to this problem in different ways.

In [8] a random factor is added as a delay. This adds some imprecision in the weight of the bow. They also describe how to use the knowledge of the probability distribution of this random variable. The authors solved this problem by applying a strategy similar to the minmax regret, using the supreme of the probability distribution of the delay.

In [9], the authors modeled robustness using confidence intervals. Although they tackled the mono-objective problem, they transformed it into a multi-objective problem by moving from exact values to confidence intervals.

Other authors have started off from the existence of a finite set of possible combinations of values that can take the weights of the arcs. These sets are called scenarios. An example of this type of robustness for the SP problem is [10]. In this paper, the authors used a minmax regret strategy to find a solution that minimizes cost in the worst case scenario.

There are many papers that analyze robustness in bi-objective problems from a theoretical point of view [11,12]. However, in all these articles the robustness is only applied to one single objective. The mean strategy for finding a solution is the minmax regret as in the mono-objective version of the problem.

6 Conclusions

In this paper a new model for the robust bi-objective shortest path problem has been presented. Our purely multi-objective approach is distinct from the rest of commonly used techniques. Thanks to our approach we can get a set of solutions with different degrees of robustness and quality with respect to two metrics. In addition, we have empirically illustrated our proposal comparing the results with those typical algorithms of the state-of-the-art and show how we can get more robust solutions than them.

As a future work, we will use real data instead of random weights and different maps for solve the RBSP problem and so analyze the behavior of our model over different set of data. Besides, we will compare our approach with another models and implemented different algorithms to resolve the BSP and RBSP problem. Also, we will apply this methodology of dealing with the robustness to other problems in smart cities. We will also try not only to treat robustness, but also to quantify its degree of use in different algorithms and applications.

Acknowledgements. This research was partially funded by the University of Málaga, Andalucía Tech, and the Spanish Ministry of Economy and Competitiveness and FEDER (grant TIN2014-57341-R).

References

1. Dijkstra, E.W.: A note on two problems in connexion with graphs. Numer. Math. **1**(1), 269–271 (1959)
2. Hart, P.E., Nilsson, N.J., Raphael, B.: A formal basis for the heuristic determination of minimum cost paths. IEEE Trans. Syst. Sci. Cybern. **4**(2), 100–107 (1968)
3. Cintrano, C., Stolfi, D.H., Toutouh, J., Chicano, F., Alba, E.: CTPATH: a real world system to enable green transportation by optimizing environmentaly friendly routing paths. In: Alba, E., Chicano, F., Luque, G. (eds.) Smart-CT 2016. LNCS, vol. 9704, pp. 63–75. Springer, Cham (2016). doi:10.1007/978-3-319-39595-1_7
4. Hansen, P.: Bicriterion path problems. In: Fandel, G., Gal, T. (eds.) Multiple Criteria Decision Making Theory and Application. LNE, vol. 177, pp. 109–127. Springer, Heidelberg (1980)
5. Ben-Tal, A., El Ghaoui, L., Nemirovski, A.: Robust Optimization. Princeton University Press, Princeton (2009)
6. Ide, J., Schöbel, A.: Robustness for uncertain multi-objective optimization: a survey and analysis of different concepts. OR Spectrum **38**(1), 235–271 (2016)
7. Duque, D., Lozano, L., Medaglia, A.L.: An exact method for the biobjective shortest path problem for large-scale road networks. Eur. J. Oper. Res. **242**(3), 788–797 (2015)
8. Cheng, J., Lisser, A., Letournel, M.: Distributionally robust stochastic shortest path problem. Electr. Notes Discrete Math. **36**, 511–518 (2010)
9. Hasuike, T.: Robust shortest path problem based on a confidence interval in fuzzy bicriteria decision making. Inf. Sci. **221**, 520–533 (2013)
10. Pascoal, M.M., Resende, M.: The minmax regret robust shortest path problem in a finite multi-scenario model. Appl. Math. Comput. **241**, 88–111 (2014)
11. Ehrgott, M., Ide, J., Schöbel, A.: Minmax robustness for multi-objective optimization problems. Eur. J. Oper. Res. (1), 17–31
12. Kuhn, K., Raith, A., Schmidt, M., Schöbel, A.: Bi-objective robust optimisation. Eur. J. Oper. Res. **252**(2), 418–431 (2016)