JOURNAL OF THE OPERATIONAL RESEARCH SOCIETY
https://doi.org/10.1080/01605682.2019.1639479

THE OPERATIONAL RESEARCH SOCIETY

Taylor & Francis
Taylor & Francis Group

ORIGINAL ARTICLE

Bi-objective methods for road network problems with disruptions and connecting requirements

Yipeng Huang[a], Andréa Cynthia Santos[a] and Christophe Duhamel[b]

[a]ICD-LOSI, Université de Technologie de Troyes, Troyes, France; [b]Université Clermont Auvergne, LIMOS, UMR CNRS 6158, Clermont-Ferrand, France

ABSTRACT

Disruptions in urban roads can significantly alter the quality of the transportation network by generating more congestion, gas emission, noise, stress, etc. In some situations, it can even break the path between some pairs of nodes in the road network (strong connectivity in graph theory). To avoid this issue, traffic managers can temporarily change the orientation of some streets (arc reversal). In this study, we propose bi-objective methods for solving the bi-objective Unidirectional and bi-objective Multidirectional Road Network problems with Disruptions and connecting requirements (resp. bi-URND and bi-MRND). In bi-URND, the road network represents local networks such as city centers with narrow streets. In this case, a simple graph is used to model the transportation network. A more general urban network is addressed with bi-MRND by means of a multi-graph model. We propose an ϵ-constraint method to compute the Pareto-optimal fronts, using an up-to-date mathematical formulation and an NSGA-II. Both bi-objective methods are compared with two metaheuristics (a Biased Random Key Genetic Algorithm and an Iterated Local Search) proposed by Huang, Santos, and Duhamel [Huang, Y., Santos, A. C., & Duhamel, C. (2018). Methods for solving road network problems with disruptions. Electronic Notes in Discrete Mathematics, 64, 175–184. 8th International Network Optimization Conference - INOC 2017] and, including an aggregation of the two objective functions. Results are presented for simulated and realistic instances on Troyes city in France.

ARTICLE HISTORY
Received 12 February 2019
Accepted 29 June 2019

KEYWORDS
Road network design; smart mobility; bi-objective optimisation; ϵ-constraint method; NSGA-II; bi-objective metaheuristics

1. Introduction

Most cities in the world are facing difficulties in managing and improving their urban network infrastructure. The impact on the daily life will become more and more noticeable due to the concentration of population in urban areas, which is already higher than the rural one. The situation turns even more difficult whenever disruptions happen on the road network, either predictable (road works, maintenance, social events, etc.) or not (accidents, bad weather conditions, disasters, etc.). They reduce the traffic capacity and pass-rate in involved areas, causing congestion and possibly breaking travel paths between some locations (loss of strong connectivity). Traffic managers can change the orientation of some streets (arc reversal), modifying thus the vehicles flow in order to maintain the network strong connectivity.

In this study, urban network management after disruptions, addressing strong connectivity and arcs reversals, is modelled and treated as bi-objective Unidirectional and bi-objective Multidirectional Road Network problems with Disruptions and connecting requirements (resp. bi-URND and bi-MRND). The former often applies to road networks in city centres, where almost all streets are one-way and the network can hence be modelled as a simple digraph. The latter corresponds to more general road networks since it uses a multi-graph, *i.e.* a graph with several arcs between any pair of nodes, where streets can be bi-directional and have multiple lanes in each direction. A disruption on the road network is defined as an unavailable arc in a digraph. The following two objectives are considered for both bi-URND and bi-MRND: minimising the total travel distance between all points of the network and minimising the number of arcs reversals. The former aims at evaluating the global efficiency of the network in terms of travel distance, while the latter aims at reducing the inconvenience for the drivers since it can disturb users habits.

Given a set B of disruptions and a simple loopless bridgless connected digraph $G = (N, A)$, where N is the set of nodes representing road intersections, and A is the set of arcs standing for road segments. Let us consider the auxiliary graph $H = (N, A')$, with $A' = A \setminus B$, which corresponds to the network of available roads for the traffic. Note that H may not be strongly connected anymore. For the sake of clarity, graph orientation means setting a direction for

CONTACT Andréa Cynthia Santos andrea.duhamel@utt.fr

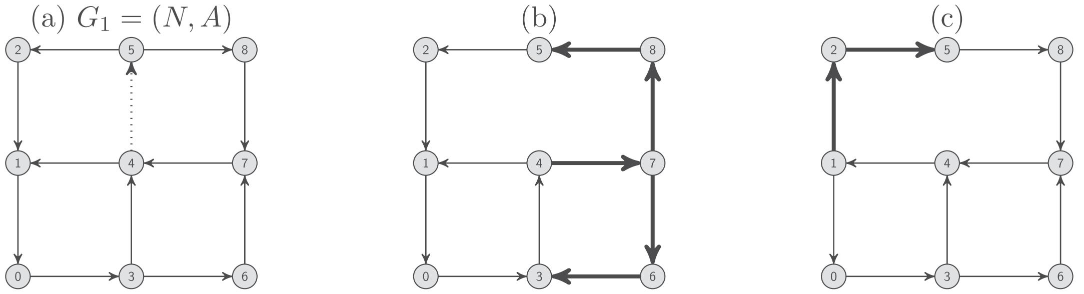


Figure 1. An example of a network with one blockage for URND.

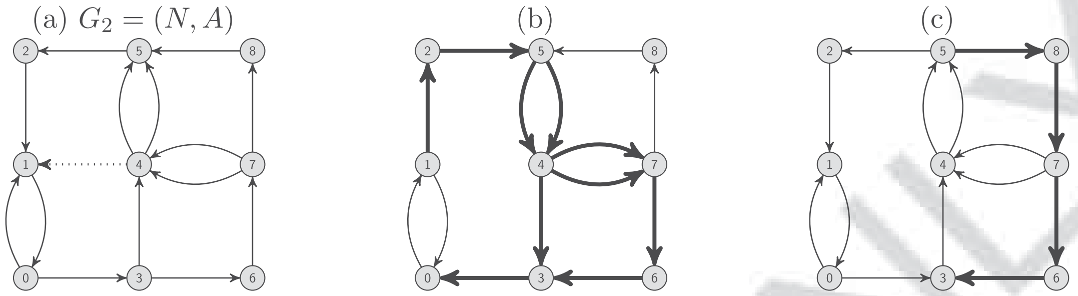


Figure 2. An example of a network with one blockage for MRND.

each edge of the graph. The bi-URND aims at defining a new orientation for H in terms of arcs orientation, which minimises simultaneously the total distance between every pair of nodes $(i, j) \in N$ and the number of arcs reversals, and such that the final graph configuration is strongly connected. Figure 1 depicts an example of different orientations on an unidirectional road network with nine nodes $\{0, 1, 2, 3, 4, 5, 6, 7, 8\}$ and one blocked arc (4, 5). Suppose G_1 models this network with the blockage illustrated in Figure 1(a). It is no longer strongly connected due to the blocked arc. Figures 1-(b) and 1-(c) are two possible feasible orientations and arc reversals are shown in bold. More precisely, a deviation for the blockage (4, 5) can go through (4, 7, 8) and (5, 8) in the orientation of Figure 1-(b), while another deviation is applied in Figure 1(c) via (1, 2, 4) and (2, 5).

A similar notation can be used for the bi-MRND in terms of graph. In this case, G is a connected loopless bridgeless directed multigraph, where A is the set of multiple arcs corresponding to lanes, either one-way or two-way streets and $A' = A \setminus B$ is the set of available arcs after disruptions. The bi-MRND is a direct generalisation of the bi-URND, since a more general network is addressed. The objective functions and constraints remain similar, but the model is more complex since it is set on a multigraph. Thus, the model and methods need to be adapted to address the generalisation to multi-arcs. Illustrative orientations are given in Figure 2. The network contains nine nodes $\{0, 1, 2, 3, 4, 5, 6, 7, 8\}$ and a single disruption (1, 4). Suppose G_2 models this network with the disruption as depicted in Figure 2(a). Two feasible orientations are shown in Figures 2-(b) and 2-(c) and arc reversals are shown in bold. Bypasses of the blocked arc (1, 4) can be done by the path (3, 4), (3, 0) and (0, 1) in Figure 2(b) and by the path (2, 4, 5) and (1, 2) in Figure 2(c).

Regarding the complexity of bi-URND and bi-MRND, orienting a graph such that the total distance between all-pairs of nodes is minimised is NP-hard, see Chvátal and Thomassen (1978), while orienting a graph such that arcs reversals is minimised and the graph is strongly connected is polynomial, see Bang-Jensen and Gutin (2008). As a consequence bi-URND and bi-MRND are NP-hard.

Several works in the literature focus on the fundamental characteristics of bi-URND and bi-MRND such as the strong connectivity and the arc reversals, as well as closely related problems like orientation problems. Robbins (1939) proposes the seminal theorem on strong orientation: a simple graph admits a strong orientation iff it does not contain a bridge. Boesch and Tindell (1980) extend this theorem for mixed multi-graphs, *i.e.* graphs that combine multi-arcs and edges. Ádám (1963) states the conjecture that there always exists an arc in a digraph whose reversal decreases the number of elementary cycles.

Considering works about setting strong orientation in graphs, Roberts (1978) proposes to compute a strong orientation on simple graphs by means of a Depth-First Search (DFS)-based algorithm. Conte, Grossi, Marino, Rizzi, and Versari (2016) propose an algorithm for enumerating all strong orientations in $\mathcal{O}(m)$ amortised time each for a given mixed multi-graph with m edges. The closely related work of Santos, Duhamel, and Prins (2013) considers the Strong Network Orientation Problem (SNOP): it aims at setting a strong orientation for an undirected graph minimising the total distance of the trips, depending on a given traffic demand. SNOP does not take into account the notion of disruptions and arcs reversals, on the contrary of bi-URND and bi-MRND. Moreover, two objectives are considered simultaneously in these two problems, as previously mentioned. This especially holds in decision making systems where many criteria are often considered simultaneously. The number of studies dedicated to multiobjective optimisation is very large. Thus, we provide the following entry points: Oliveira and Ferreira (2000), Ehrgott and Gandibleux (2000), Junior and Lins (2009), Zopounidis and Pardalos (2010) and Zhou, Qu, Li, Zhao, Suganthan, and Zhang (2011) for survey and books closely related to network design problem; and, Santos, Lima, and Aloise (2014) and De Sousa, Santos, and Aloise (2015) for bi-objective network design applications.

The following previous studies on URND and MRND were done, in chronological order. A first mathematical model for bi-URND is proposed in Huang, Santos, and Duhamel (2016a), and it is solved using a commercial Mixed Integer Linear Programming solver (MILP). The model was tested over small grid graph instances with up to 36 nodes and 60 arcs. Pareto-fronts were computed to up to 25 nodes. Graph theoretical issues were handled in Huang, Santos, and Duhamel (2016b) in order to simplify the use of multi-graphs and, as a consequence, be able to handle more general urban networks. We showed that for the two objectives considered in bi-MRND, any multi-graph can be reduced to a 2-directed multi-graph, i.e. a multigraph with at most two arcs between any two nodes. Scheduling disruptions in a multi-period time horizon is done in Coco, Duhamel, and Santos (2019). In Huang et al (2018), a Biased Random Key Genetic Algorithms (BRKGA), see Gonçalves and Resende (2011); Resende (2012), and an Iterated Local Search, see Lourenço, Martin, and Stützle (2010), were developed. For the former, the Pareto front is obtained by extracting all non-dominated solutions found in the iterations of BRKGA. For the latter, an aggregation of the two objective functions is done. More recently in Huang, Santos, and Duhamel (2019), an improved mathematical formulation is proposed for URND and MRND. The two objectives were addressed separately using a BRKGA and an ILS.

This work extends our previous studies Huang et al (2018, 2019), by applying efficient multiobjective methods. Precisely, the improved model of Huang et al (2019) is used in an ϵ-constraint scheme and an NSGA-II (Deb, Pratap, Agarwal, and Meyarivan (2002)) is adapted to the two problems. The two metaheuristics BRKGA and ILS previously applied in Huang et al (2019) are adapted to the multiobjective context by using an aggregated objective function. The goal is to analyse the performance of classical multi-objective methods with the adaptation of BRKGA and ILS. Moreover, all the methods are evaluated using multi-objective indicators like hypervolume, spread and number of points in the Pareto-front.

It is worth mentioning that the formalisation of URND and MRND in theoretical terms and the proposed of methods help the overall understanding of a relevant problem happening in an urban context. These problems come from an undergoing collaborative project with the traffic management centre of Troyes city in France.[1] The software enclosing our methods was awarded a "Le Monde Smart City" European Prize of Innovation in the Urban Mobility category in 2017.

The remaining of this article is organised as follows. Initially, in Section 2, the ϵ-constraint exact method is presented, together with the improved model. Then, Section 3 is dedicated to the bi-objective metaheuristics, where the bi-objective ILS and the bi-objective BRKGA are detailed, followed by the description of the NSGA-II. Numerical results are presented in Section 4, before concluding remarks in Section 5.

2. An ϵ-constraint exact method

Solving a multiobjective optimisation problem implies finding a set of solutions in the search space using usually the Pareto concept proposed by Pareto (1906). Let us S be the set of all feasible solutions in a search space for a multiobjective problem. Given two solutions $s_1 \in S$ and $s_2 \in S$ in a minimisation problem, where s_1 dominates s_2, if for every criterion z_i, the cost $z_i(s_1) \leq z_i(s_2)$, and there is at least one criterion z_i such that $z_i(s_1) < z_i(s_2)$. The Pareto-optimal front hence contains the nondominated solutions (Chankong and Haimes, 1983).

For the sake of clarity, let us consider $\mathcal{F}$ a Pareto front and $\mathcal{F}^*$, an optimal Pareto-front. The ϵ-constraint method is based on the enumeration strategy proposed by Haimes, Lasdon, and Wismer (1971). This method relies on a precision parameter $\epsilon>0$, which is used to iterate on the criterion turned into a parametric constraint. When this criterion takes integer values, ϵ can be set to 1. This means the search space is constrained by successively optimising one of the objectives by means of a mathematical model or a heuristic. The corresponding solution value will be used to set the new bound. The procedure stops when the sub-problem becomes infeasible. In some cases, the extreme

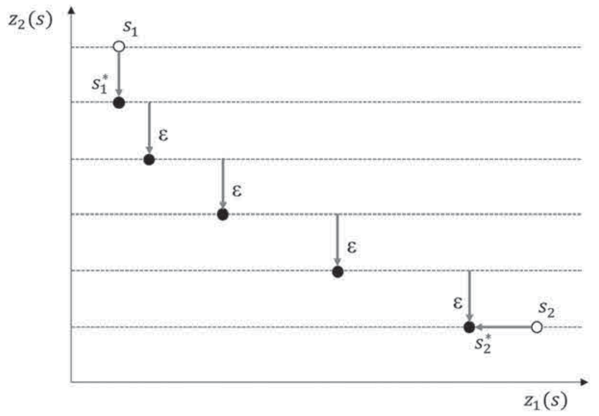


Figure 3. General schema of ϵ-constraint method.

optimal solutions of the Pareto-front can be computed and used as a stopping criterion. It is worth mentioning that the Pareto optimal front $\mathcal{F}^*$ is built, whenever an exact method is used at each step of the ϵ-constraint method. Examples of applications for the ϵ-constraint method are found in Bérubé, Gendreau, and Potvin (2009) and De Sousa et al (2015).

Considering a bi-objective minimisation problem, Figure 3 illustrates the way the ϵ-constraint method iteratively computes the Pareto-optimal solutions. Solutions s_1 and s_2 are first computed by performing a single-objective optimisation on z_1 and z_2 criteria, respectively. Then, the extreme point s_1^* is obtained by optimising z_2 criterion with the additional constraint $z_1(s) = z_1(s_1^*)$. Then, by iteratively reducing the upper-bound constraint on $z_2(s)$, the sub-problem computes all the Pareto-optimal solutions until the extreme solution s_2^* is found.

2.1. Mathematical formulation for bi-URND and bi-MRND

The mathematical formulation used in the proposed *ϵ-constraint* method for bi-URND is defined from (1) to (8). It makes use of the graph H previously defined. In addition, a cost θ_{ij}, corresponding to the distance, is associated with every arc $(i,j) \in A'$. Let $P = \{(o,d) | o,d \in N, o \neq d\}$ be the set of all (O)rigin-(D)estination pairs in H. Since H is obtained from G by removing the disrupted arcs, it may not be strongly connected anymore. In such a case, the strong connectivity must be restored by reversing arcs and any arc in A' can be reversed. Moreover, the strong connectivity of the new orientation is ensured by sending a unit flow for each OD pair $(o,d) \in P$. Furthermore, the proposed model uses boolean variables $x_{ij} \in \{0,1\}$ indicating whether arc $(i,j) \in A'$ is reversed $(x_{ij} = 1)$ or not $(x_{ij} = 0)$. Variables $f_{ij}^{od} \in \mathrm{IR}$ define the flow from o to d on the arc (i, j), taking into account its final orientation. Assuming the arc (i, j) is used to send flow from o to d, $f_{ij}^{od}>0$ if the arc is not reversed and $f_{ij}^{od} < 0$ otherwise. In this way, flow conservation constraints hold on variables f. Variables $y_{ij}^{od} \geq 0$ define the amount of flow from o to d going through arc (i, j). Thus, they do not depend on the orientation of the arc (i, j).

The bi-objective function is given in (1) for minimising both the total travelling distance and the number of arcs reversals. The flow conservation constraints are shown in (2) and (3), using variables f_{ij}^{od}. Since the variables f_{ij}^{od} take both the amount of flow and the reversal state into account, the relation between variables f_{ij}^{od}, y_{ij}^{od} and x_{ij} are defined in inequalities (4) and (5), which are a linearisation of the nonlinear Equation (9). Variables are defined from (6) to (8). This model contains $\mathcal{O}(|P| \times |A'|)$ constraints and variables.

$$\text{minimize} \begin{cases} c_1(y) = \sum_{(o,d)\in P} \sum_{(i,j)\in A'} \theta_{ij} y_{ij}^{od} \\ c_2(x) = \sum_{(i,j)\in A'} x_{ij} \end{cases} \tag{1}$$

s.t.

$$\sum_{l:(l,o)\in A'} f_{lo}^{od} - \sum_{l:(o,l)\in A'} f_{ol}^{od} = -1 \quad \forall (o,d) \in P, \tag{2}$$

$$\sum_{l:(l,i)\in A'} f_{li}^{od} - \sum_{l:(i,l)\in A'} f_{il}^{od} = 0 \quad \forall (o,d) \in P, \forall i \notin \{o,d\}, \tag{3}$$

$$-y_{ij}^{od} \leq f_{ij}^{od} \leq y_{ij}^{od} \quad \forall (o,d) \in P, \forall (i,j) \in A', \tag{4}$$

$$-2x_{ij} + y_{ij}^{od} \leq f_{ij}^{od} \leq 2-2x_{ij}-y_{ij}^{od} \quad \forall (o,d) \in P, \forall (i,j) \in A', \tag{5}$$

$$x_{ij} \in \{0,1\} \quad \forall (i,j) \in A', \tag{6}$$

$$f_{ij}^{od} \in \mathrm{IR} \quad \forall (o,d) \in P, \ \forall (i,j) \in A', \tag{7}$$

$$y_{ij}^{od} \geq 0 \quad \forall (o,d) \in P, \ \forall (i,j) \in A'. \tag{8}$$

This mathematical model can be adapted to bi-MRND by taking the lanes into account. It is sufficient to set an index for each arc $(i,j) \in A'$ such that a multiple arc $(i,j) \in A'$ corresponds to K copies (i, j), each with a different index, starting from 0 and going up to K – 1. This way, each arc in A' is uniquely defined by a triple (i, j, k), i being the origin, j the destination and k the index. Then, variables x_{ij} become x_{ijk}. Same for f_{ij}^{od} and y_{ij}^{od} which become respectively f_{ijk}^{od} and y_{ijk}^{od}. The adaptation is straightforward and does not alter the structure of the formulation.

$$f_{ij}^{od} = y_{ij}^{od} \times (1-2x_{ij}) \quad \forall (o,d) \in P, \forall (i,j) \in A'. \tag{9}$$

2.2. ε-constraint applied to bi-URND and bi-MRND

Given a solution $s \in S$ with respect to bi-URND or bi-MRND. Let $x(s)$ and $y(s)$ define the x and y

variables associated with s, respectively. The dominance rule is defined as follows. Considering two solutions $s_1 \in S$ and $s_2 \in S$, s_1 dominates s_2 ($s_1 \prec s_2$) either if (i) $c_1(y(s_1)) < c_1(y(s_2))$ and $c_2(x(s_1)) \leq c_2(x(s_2))$ or if (ii) $c_1(y(s_1)) \leq c_1(y(s_2))$ and $c_2(x(s_1)) < c_2(x(s_2))$. Then s_2 is nondominated if no such s_1 exists, *i.e.* $\nexists s \in S | s \prec s_2$.

In the ϵ-contraint, the computation of the Pareto-optimal front $\mathcal{F}^*$ for bi-URND starts by first computing the extreme solitions s_1^* and s_2^* associated with the optimal non-dominated solutions for $c_1(y)$ and $c_2(x)$ criteria. This is done by solving successive Mixed-Integer Linear Programs (MILP). The values u_2 and l_2 stand respectively for the number of arc reversals in s_1^* and s_2^*. Thus, $u_2 - l_2$ defines the interval of variation in $\mathcal{F}^*$ for the ϵ-constraint method. Since $c_2(x(s)) \in \mathbb{N}$, setting $c_2(x)$ as a parametric constraint in the method with $\epsilon = 1$ ensures all solutions of $\mathcal{F}^*$ will be found.

Given κ the current value on c_2 criterion, the subproblem to be solved is given in Equation (10), where S is the domain of feasible solutions defined by constraints (2)–(8). In bi-URND, the bounding constraint $c_2(x(s)) \leq \kappa$ corresponds to Equation (11).

$$\left\{\begin{array}{l} \min c_1(y(s))\ \ s.t. \\ c_2(x(s)) \leq \kappa \\ s \in S \end{array}\right\} \tag{10}$$

$$\sum_{(i,j)\in A'} x_{ij} \leq \kappa \tag{11}$$

The same approach is used for bi-MRND except that arcs are defined by a triple (i, j, k) where i and j are, respectively, the origin and the destination, and k is the index among all the arcs from i to j. Thus, inequality (12) is used for bi-MRND instead of (11).

$$\sum_{(i,j,k)\in A'} x_{ijk} \leq \kappa \tag{12}$$

Details of this adapted method are given in Algorithm 1. The initialisation of the method is done in lines 1–5, where the extreme Pareto-front solution s_1^* (line 2), the Pareto optimal front $\mathcal{F}^*$ (line 4), and κ (line 5) are set. Loop 6 to 10 corresponds to reducing κ iteratively until all non-dominated solutions between s_1^* and s_2^* are obtained. The optimal Pareto-front is returned in line 11.

3. Bi-objective metaheuristics for bi-URND and bi-MRND

We propose three heuristics to handle the two objective functions focused in bi-URND and bi-MRND. ILS and BRKGA are originally designed for single-objective problems. However, it can be easily adapted to address bi-objective problems, using an aggregated function. The reason for adapting such methods is due to their efficiency on single-objective applications, and also since the modifications remain simple and can be done without deteriorating their main advantages. In addition, one of the more successful metaheuristics for multi-objective problems, **NSGA-II** has also been developed. These methods are detailed in this section.

Algorithm 1 ϵ*−constraint*(G, ϵ)

```
1: Input
2:     $H$ graph (simple or multiple) with initial orientation
3:     $\epsilon$ precision
4: Output
5:     $\mathcal{F}^*$ Pareto optimal front of $G$
6: $s_1 \leftarrow \min\{c_1(y(s)) | s \in S\}$
7: $s_1^* \leftarrow \min\{c_2(x(s)) | c_1(y(s)) = c_1(y(s_1)), s \in S\}$
8: $s_2 \leftarrow \min\{c_2(x(s)) | s \in S\}$
9: $\mathcal{F}^* \leftarrow \{s_1^*\}$
10: $\kappa \leftarrow c_2(x(s_1^*))$
11: repeat
12:     $s \leftarrow \min\{c_1(y(s)) | c_2(x(s)) \leq \kappa - \epsilon, s \in S\}$
13:     $\mathcal{F}^* \leftarrow \mathcal{F}^* \cup \{s\}$
14:     $\kappa \leftarrow c_2(x(s))$
15: until $\kappa = c_2(x(s_2))$
16: return $\mathcal{F}^*$
```

3.1. ILS adapted to bi-URND and bi-MRND

ILS has been proposed by Lourenço, Martin, and Stützle (2003) and several successful applications have been presented in Lourenço et al (2010). Differing from population-based algorithms like BRKGA, this method improves a single feasible initial solution by iteratively applying a local search followed by a perturbation. When a predefined stopping criterion is reached, ILS outputs the incumbent solution found during the process. The method contains three major components: a constructive algorithm to generate the initial feasible solution, a local search procedure and a shaking operator for the perturbation.

An aggregate objective function using a weighted-sum approach is defined as in Equation (13), in order to handle bi-URND and bi-MRND using the ILS, referred to as bi-ILS. Given a solution s, a weighting parameter $\alpha \in (0, 1)$, and the objective values of $c_1(y(s))$ and $c_2(x(s))$. The objective functions are evaluated and normalised, and then aggregated into one objective value $f_\alpha(s)$. The parameter α is set in the interval (0, 1). One may note that if $\alpha = 0$, only $c_2(x(s))$ is optimised. On the contrary, if $\alpha = 1$, the optimisation is performed for $c_1(y(s))$.

$$\min_{s\in S} f_\alpha(s) = \alpha c_1(y(s)) + (1-\alpha) c_2(x(s)) \quad \alpha \in (0, 1) \tag{13}$$

The construction of a feasible initial solution is done as follows. Initially, a random weight is assigned to every node the graph H. Then, a DFS-based strong orientation algorithm is applied. As shown in Roberts (1978), a DFS algorithm can be used to generate strong orientation for graphs. The weights allow to generate different orientations. The weights are used to define the order of node visits in the DFS. First, the initial orientation of the given graph is removed such that every edge has to be oriented. At each step of the algorithm, given the current node $i \in H'$, the set R_i of reachable neighbour nodes, i.e. all the nodes that are connected to i by an edge $[i,j]$, is built. Then, the node $j \in R_i$ with the largest weight is selected, the edge $[i,j]$ is oriented from i to j and the method iterates.

The local search used in ILS is the Variable Neighbourhood Descent (VND) developped by Mladenović and Hansen (1997). In our VND, the current solution is improved by iteratively moving to the next neighbourhood using a first improvement strategy. The search stops whenever no improvement is found. Three neighbourhood structures $\mathcal{N}_k, k = 1...3$, are used in the VND. A move in $\mathcal{N}_1$ consists in reversing all the arcs of a cycle. In $\mathcal{N}_2$, all the arcs incident to a node are reversed and a single arc is reversed in $\mathcal{N}_3$. Moves in $\mathcal{N}_1$ do not break the strong connectivity, on the contrary of $\mathcal{N}_2$ and $\mathcal{N}_3$. Thus, for these last two neighbourhoods, only strong connected neighbour solutions are considered. Finally, two shaking functions are developed for the perturbation. Given the current solution, the first shaking reverses the direction of a cycle while the second shaking changes the directions of all the arcs. The first one is used in the first half of ILS iterations and the second one is used in the second half.

The Pareto front $\mathcal{F}$ is held by a meta-process that runs bi-ILS with different values of α distributed uniformly in (0, 1). During each run, $\mathcal{F}$ is updated every time an improving solution s is found: (i) s is inserted into $\mathcal{F}$ if it is nondominated in $\mathcal{F}$ and (ii) all the dominated solutions $\mathcal{F}$, *i.e.* $\{s' \in \mathcal{F} | s \prec s'\}$ are removed from $\mathcal{F}$. A post-processing step is added to each run of bi-ILS. It tries to iteratively reverse back the reversed arcs, while the orientation remains strongly connected. This step is used to reduce as much as possible the value $c_2(x(s))$ since the tree structures $\mathcal{N}_1$,$\mathcal{N}_2$ and $\mathcal{N}_3$ are not designed towards $c_2(x(s))$ optimisation.

3.2. BRKGA adapted to bi-URND and bi-MRND

BRKGA has been proposed by Gonçalves and Resende (2011) and Resende (2012). It is an extension of Random Key Genetic Algorithm (RKGA) from Bean (1994), a variant of GA. Both BRKGA ans RKGA use a vector of random keys in $[0,1]$ to represent an individual in their population. Thus, a decoder is used in order to transform this vector into a feasible solution, depending on the specification of the given problem. In this way, the problem-specific parts are kept in the decoder, hence isolated from the algorithmic scheme of BRKGA.

In our version of BRKGA for bi-URND and bi-MRND, refereed to as bi-BRKGA, random keys are set as a vector of weight associated with each node of the given graph H. Then, the decoder works as in the ILS, by using an adapted version of the DFS to compute a strong orientation. The highest weights impose the orientation of arcs. During a run of BRKGA, the criterion $c_1(y(s))$ is evaluated by computing the shortest path between all pairs of nodes. This step is done using the classical shortest path algorithm of Dijkstra, see Cormen, Leiserson, Rivest, and Stein (2009). The second criterion $c_2(x(s))$, *i.e.* the number of arcs reversals, is computed by counting the number of arcs whose direction differs from their initial orientation. The aggregated function $f_\alpha(s)$ (see Equation 13) is applied and the Pareto front $\mathcal{F}$ evolves along with the iterations of the BRKGA. Every time a so-far nondominated solution s is found, $\mathcal{F}$ is updated in a way that (i) s is inserted into $\mathcal{F}$ and (ii) all the dominated solutions in $\mathcal{N}$,*i.e.* $\{s' \in \mathcal{F} | s \prec s'\}$ are removed from $\mathcal{F}$.

3.3. NSGA-II

Multi-objective evolutionary algorithms (MOEA) are an effective way to compute good approximations of the Pareto-optimal front. Among them, adaptations of GA have led to efficient algorithms such as the Nondominated Sorting Genetic Algorithm (NSGA) proposed by Srinivas and Deb (1994), based on the ranking mechanism. In addition, a sharing method with a technique is applied in order to keep diversity in the population and in the distribution of nondominated solutions. The other mechanisms of the metaheuristic, such as the crossover and the mutation, follow the guidelines of a classical GA.

Some drawbacks of NSGA have been identified by the community, including the high complexity due to the implementation of the nondominated sorting, the lack of an elitist mechanism and the dependence on the sharing parameter. In Deb et al (2002), these disadvantages are well explained. They are addressed in an enhanced approach, NSGA-II. This improved method integrates a fast nondominated sorting with a lower complexity. The main idea is, for every solution, to get the number of solutions dominating others and to store all the solutions dominated by others. The time complexity is proved to be $\mathcal{O}(MN^2)$ where M is the number of criteria and N is the population size. Yet, this new approach requires a slightly higher storage space in $\mathcal{O}(N^2)$. NSGA-II also replaces the sharing function with a crowded-comparison operator that improves the solutions distribution on the Pareto front as well. Instead of sharing the fitness, a crowding distance is computed for every solution to indicate whether it is located in a crowded region or not of the current rank. The selection of the GA is then guided by comparing the crowding distances in a way that solutions in less crowded areas are preferred.

The proposed NSGA-II for bi-URND and bi-MRND uses the generic fast nondominated sort with the ranking calculation and the crowding distance. Two encodings have been developed, one based on weights associated with nodes and another using weights associated with arcs. The first one is similar to the idea used in the BRKGA, where the vector of weights associated with nodes is used to generate strong orientation by means of a DFS algorithm. In the second encoding, a weight $w_{ij} \in (0, 1)$ is associated with each arc $(i, j) \in A'$. Then, if $w_{ij} < 0.5$, the edge $[i, j]$ is oriented from the node with the lower index to the node with the higher index, and from the higher index to the lower index otherwise ($w_{ij} \geq 0.5$). The first encoding was abandoned since it leads to a lower performance in a **NSGA-II** scheme.

One may note that generating a strong orientation using arcs does not ensure feasibility. Thus, a penalisation is added in order to remove the need for a repair function and to avoid the bias any reparation could induce. The penalisation consists in adding a large-enough predefined value each time an OD pair is not connected. This way, infeasible solutions are kept, but they are dominated automatically by every feasible solution with the same number of arc reversals. Besides, they are removed from the final Pareto front at the end of the run. The algorithm used by this decoding is shown in Algorithm 2. Its complexity is trivially $\mathcal{O}(m)$ with m being the number of arcs.

Algorithm 2 *ArcOrient*(G, w)

```
Input
    $H$ graph (simple or multiple) to be oriented
    $w$ weight vector on arcs
Output
    $H$ with an orientation
for each $[i, j] \in A'$ do
    $a \leftarrow$ its node with smaller index
    $b \leftarrow$ its node with larger index
    if $w_{ij} < 0.5$ then
        orient $[i, j]$ from $a$ to $b$
    else
        orient $[i, j]$ from $b$ to $a$
    end if
end for
```

The crossover applied in the NSGA-II is the one proposed by Spears and Jong (1991), named parametrised uniform crossover. In this crossover, one individual is always selected from the elite set, while the other is randomly chosen in the population. The probability of a solution inheriting the allele from the elite individual is equal to 0.7 as suggested by Gonçalves and Resende (2011). The number of mutants and the elite set size follow the work of Gonçalves and Resende (2011), where the recommendation is to include a small number of mutants into the population at each generation. This is set to 10% of the overall population. In Gonçalves and Resende (2011), experiments indicate elite sets ranging from 15% to 25% of the overall population tend to produce a better performance of BRKGA. Here, it is set to 20%.

4. Numerical results

Numerical experiments were addressed to evaluate the computational performance of the proposed exact method and the three bi-objective metaheuristics, as well as their limits in terms of solution quality and running time. All the experiments are performed on a machine with an Intel Core i7-4710MQ CPU at 2.50 GHz with 4 cores, 16 GB RAM under Windows 7. CPLEX version 12.6 is used with default parameters and a time limit of 2 h is set. All the procedures are coded in C++ and compiled by Visual C++ version 11.0 (Visual Studio 2012).

Simulated and realistic instances from Troyes city in France are used in the experiments. The simulated instances are grid graphs with a unit arc cost. They include simple graph instances for bi-URND and multigraph instances for bi-MRND. For the latter, multiple arcs are generated randomly such that no more than 2 arcs exist between any pair of nodes. In every instance, disruptions are selected on a random basis and the number of disruptions varies in $\{1, 2, 4, 6\}$. In addition, when generating disruptions, we ensure that the resulting graph can always admit a strong orientation, such that feasible solutions should exist.

The realistic instances were generated for bi-MRND as follows. We extracted geographical data from the Troyes urban network, a city in France with about 61,000 inhabitants in 2015, using OpenStreetMap [2]. The original dataset from OSM

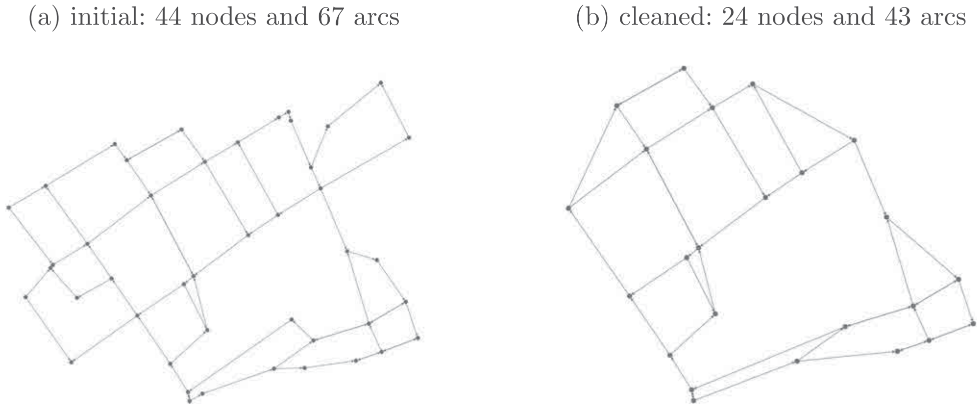

(a) initial: 44 nodes and 67 arcs
(b) cleaned: 24 nodes and 43 arcs

Figure 4. Troyes partial downtown road network modelled by graphs.

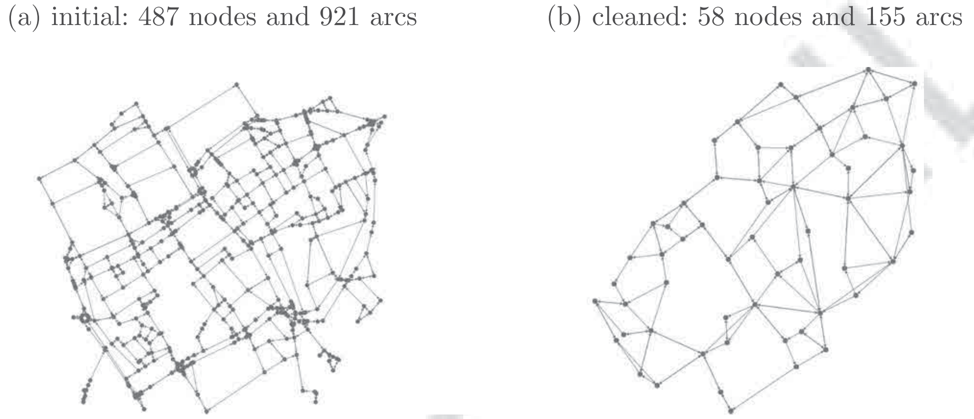

(a) initial: 487 nodes and 921 arcs
(b) cleaned: 58 nodes and 155 arcs

Figure 5. Troyes downtown road network modelled by graphs.

consists of 4261 nodes and 614 ways including useless information in our context like buildings, squares and internal ways in parking lots. A data preprocessing procedure is applied in order to build bridgeless graphs. Two graphs are used: one with 24 nodes and 43 arcs and another one with 58 nodes and 155 arcs. Four scenarios of disruptions are applied for each graph, resulting in 8 instances, named "troyes-vδ-λ", where δ and λ stand respectively for the number of nodes and the number of disruptions. Disruptions are randomly selected such that the resulting graph does not contain bridges. Figures 4 and 5 depict the addressed network modelled by multigraphs. Figures 4(a) and 5(a) illustrate the initial data, while Figures 4(b) and 5(b) show the graphs resulting from the preprocessing phase.

Preliminary experiments on the parameters' values were performed, followed by three experiments. In the first experiment, we have tested the ϵ-constraint method and the three metaheuristic methods (NSGA-II, bi-BRKGA and bi-ILS) on small simulated instances. In the second one, larger instances are solved by the metaheuristics, since the exact method can no longer handle such sizes. Third, the metaheuristics are tested on instances corresponding to the road network of Troyes.

To analyse the quality of Pareto fronts produced by the proposed methods, three multiobjective metrics are used. The first metric is the number of Pareto solutions in the first front. The second metric is the hypervolume (Zitzler and Thiele, 1998) that represents the size of the area dominated by the given front in the solution space. The last one is the spacing (Veldhuizen and Lamont (2000) and Zitzler, Thiele, Laumanns, Fonseca, and da Fonseca (2003)) which allows to analyse the distribution of the solutions on the given Pareto front. In all tables, values in bold correspond to the best results.

In the preliminary experiments, the following parameters were tested for each method. The multiplier on the number of nodes (R_N) used to compute the population size, the maximal number of generations (I_N), the probability of crossover (ρ_{cN}) and the probability of mutation (ρ_{mN}) were tested for the NSGA-II. In bi-BRKGA, the key parameters are the multiplier on the number of nodes (R_B) used to set the number of individuals in the population and the number of generations without profit (J_B). They correspond to the population size and the stopping criterion. In bi-ILS, the parameters are the number of iterations without profit (J_I) and the rate of variation on the weight parameter α, refer to as (Δ_I).

For the sake of reproducibility and according to the results from preliminary tests, the sets of chosen parameter values for each metaheuristic are reported in Table 1.

4.1. Results on small instances

Tables 2 and 3 report, respectively, the results for bi-URND and bi-MRND obtained on the instances with up to 25 nodes. The columns "*Q*", "*HV*", "*SP*" and "t(s)" present, respectively, the number of solutions in the Pareto front, the hypervolume (the larger the better), the spacing (the smaller the better) and the computational time in seconds. Note that the objective values are normalised in the computation of spacing, since the values of $c_1(y(s))$ and $c_2(x(s))$ are not in the same scale.

Results in Table 2 indicate that bi-ILS can compute a high quality approximation of the Pareto-optimal fronts for bi-URND (simple graph instances). This is confirmed by the tiny gap of hypervolume comparing with the Pareto-optimal fronts constructed by the ϵ-constraint method. The spacing values seem also reasonable for a non-evolutionary metaheuristic without any functions that ensure the solutions distribution. Note that most solutions found by bi-ILS on the final front are supported, which may result from the aggregation of the two criteria, whereas a few non-supported solutions are also obtained thanks to the post-processing integrated in the method. The other two metaheuristics, bi-BRKGA and NSGA-II, can construct Pareto fronts with more solutions than bi-ILS because of the use of populations, especially on 5x5 instances. The quality of their fronts are good as well, on par with bi-ILS in general. Moreover, NSGA-II computes Pareto fronts with a small spread on these instances. All the three metaheuristics can approximate the Pareto-optimal fronts in less than 2 minutes.

Table 1. Chosen parameter values.

NSGA-II				bi-BRKGA		bi-ILS	
R_N	I_N	ρ_{cN}	ρ_{mN}	R_B	J_B	J_I	Δ_I
1.75	50	0.3	0.1	1.45	6	6	0.15

The results for bi-MRND are reported in Table 3. bi-BRKGA has a good performance in terms of quality and time. It performs especially well in terms of the spread, with leading spacing values on 7 out of 8 instances. Besides, bi-ILS can still construct Pareto fronts of high quality while it becomes significantly more time-consuming than for bi-URND. The behaviour of NSGA-II is similar to the experiments on simple graph instances. It computes the Pareto fronts with a number of solutions within a reasonable time. However, starting from 6 × 6 instances, the ϵ-constraint method ran out of time (2 h time limit set to CPLEX) due to the large number of variables in the mathematical model.

Examples of Pareto(-optimal) fronts constructed by the four methods are illustrated in Figure 6 for a bi-URND instances (Figure 6(a)) and a bi-MRND instances (Figure 6(b)).

4.2. Results on large instances

The ϵ-constraint method cannot handle instances larger than 25 nodes due to the time and memory limits. Thus, for every instance with more than 25 nodes addressed in this subsection, nondominated solutions among all the solutions found by the three metaheuristics are merged in order to build a set of "Best Known Solutions (BKS)", presented within the column "*BKS*" in Tables 4 and 5. It can hence be

Table 2. Performance comparison of Pareto fronts for bi-URND (I).

Instance	ϵ-constraint			NSGA-II				bi-BRKGA				bi-ILS			
	Q	HV	t(s)	Q	HV	SP	t(s)	Q	HV	SP	t(s)	Q	HV	SP	t(s)
4 × 4-b1	2	**0.67**	**2.34**	2	**0.67**	**0.00**	24.09	3	0.07	0.46	21.65	2	**0.67**	**0.00**	8.46
4 × 4-b2	**4**	**0.50**	24.07	4	**0.50**	**0.46**	24.73	4	**0.50**	**0.46**	16.23	4	**0.50**	**0.46**	**6.30**
4 × 4-b4	**5**	**0.72**	11.59	4	0.71	0.89	20.22	5	**0.72**	0.83	8.38	4	0.63	**0.82**	**4.49**
5 × 5-b1	**15**	**0.89**	3769.28	11	0.80	**0.94**	62.21	12	0.80	1.78	87.32	9	0.87	3.97	**42.28**
5 × 5-b2	**13**	**0.90**	1835.61	11	**0.90**	**0.65**	64.15	13	0.87	0.86	87.99	7	0.88	3.90	**30.45**
5 × 5-b4	**11**	**0.90**	1021.64	7	0.84	4.24	59.34	10	0.84	**0.59**	77.75	8	0.89	1.27	**25.69**

Table 3. Performance comparison of Pareto fronts for bi-MRND (I).

Instance	ϵ-constraint			NSGA-II				bi-BRKGA				bi-ILS			
	Q	HV	t(s)	Q	HV	SP	t(s)	Q	HV	SP	t(s)	Q	HV	SP	t(s)
4 × 4-b1-50	**8**	**0.75**	**13.85**	8	0.74	1.37	44.15	8	0.74	**0.05**	19.02	5	0.72	0.81	19.14
4 × 4-b1-100	**8**	**0.59**	**4.95**	8	**0.59**	**0.06**	68.16	8	**0.59**	**0.06**	22.08	4	0.46	1.93	16.63
4 × 4-b2-50	**7**	**0.72**	**9.22**	7	0.69	0.72	41.96	7	0.66	**0.08**	17.38	6	0.70	0.46	14.70
4 × 4-b2-100	9	**0.81**	**6.51**	10	0.76	2.81	67.52	9	**0.81**	**0.08**	23.81	4	0.75	1.78	21.54
5 × 5-b1-50	13	**0.91**	150.56	20	0.83	1.90	127.05	13	0.88	**0.63**	104.94	12	**0.91**	1.14	**76.39**
5 × 5-b1-100	21	**0.84**	101.29	26	0.78	1.27	190.65	21	0.83	**0.14**	122.09	11	0.82	1.55	**121.37**
5 × 5-b2-50	14	**0.88**	240.56	14	0.87	**0.52**	121.40	17	0.81	0.59	121.68	10	0.86	3.91	**96.56**
5 × 5-b2-100	**21**	**0.79**	135.24	21	0.72	0.92	182.46	21	0.76	**0.13**	121.68	11	0.78	1.34	**106.22**

used as the basis for the relative comparison of their performance.

Regarding the bi-URND on 7 × 7 and 8 × 8 instances, results of the three metaheuristics are shown in Table 4. bi-BRKGA and NSGA-II have globally a good performance on the hypervolume, comparing to the BKS. They can find good quality fronts with a large number of solutions. Besides, the spread performance of these two methods is satisfactory and generally better than bi-ILS. In particular, NSGA-II consumes less time than the other two methods. A possible explanation might be that NSGA-II runs with a relatively small population and less generations. On the other hand, the Pareto fronts found by bi-ILS are very close to BKS on these instances but with minor gaps in terms of number of solutions. Figure 7(a) illustrates the Pareto fronts constructed by the three metaheuristics on the 8x8 bi-URND instance with 6 disruptions.

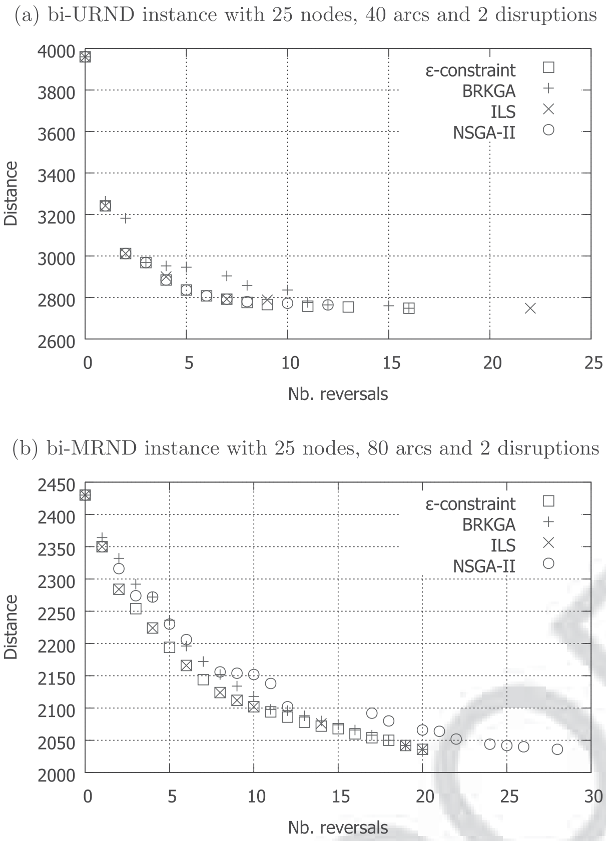


Figure 6. Pareto fronts for bi-URND and bi-MRND instances (5x5).

For bi-MRND (results reported in Table 5), bi-ILS starts to consume more time when the problem size is getting larger, particularly on the 8 × 8 instances. Sometimes, it spends more than two hours to construct a Pareto front while BRKGA and NSGA-II can finish most of the tasks in less than one hour. It seems that bi-ILS performs too many runs with different values of α, whereas the number of generations in bi-BKRGA and NSGA-II is relatively small. In terms of front quality, bi-ILS still has a significant lead over the two other metaheuristics, since the values of hypervolume on the constructed Pareto fronts are higher for all the instances presented in Table 5. In terms of solutions distribution, bi-BRKGA has a better spacing on 6 out of 8 instances than bi-ILS and NSGA-II, while bi-ILS also finds 2 fronts out of 8 instances that are better distributed than the other two methods. NSGA-II performs surprisingly worse than bi-BRKGA and bi-ILS using this metric on these instances. Figure 7(b) illustrates the Pareto fronts constructed by the three metaheuristics on the 8x8 bi-MRND instance with 1 disruption.

Table 4. Performance comparison of Pareto fronts for bi-URND (II).

Instance	BKS		NSGA-II				bi-BRKGA				bi-ILS			
	Q	HV	Q	HV	SP	t(s)	Q	HV	SP	t(s)	Q	HV	SP	t(s)
7 × 7-b1	15	**0.78**	8	0.74	**0.82**	**368**	16	0.61	1.09	1369	11	0.74	2.30	703
7 × 7-b2	19	**0.83**	20	0.58	**1.82**	**367**	17	0.76	1.75	1276	14	**0.83**	1.98	739
7 × 7-b4	7	**0.91**	15	0.83	**0.80**	**346**	13	0.75	4.13	1100	7	**0.91**	1.50	587
7 × 7-b6	14	**0.87**	13	0.72	3.70	**362**	24	0.74	**3.56**	1179	9	0.85	6.05	701
8 × 8-b1	17	**0.94**	23	0.82	4.63	**732**	33	0.87	**2.40**	4556	15	0.93	3.80	2158
8 × 8-b2	15	**0.92**	28	0.87	1.97	**716**	22	0.88	**1.76**	3474	10	0.86	5.61	2024
8 × 8-b4	18	**0.93**	21	0.83	1.79	**677**	26	0.84	**0.26**	4894	17	0.91	3.45	1898
8 × 8-b6	18	**0.94**	32	0.80	1.22	**655**	26	0.85	3.96	2840	17	**0.94**	**1.20**	1727

Table 5. Performance comparison of Pareto fronts for bi-MRND (II).

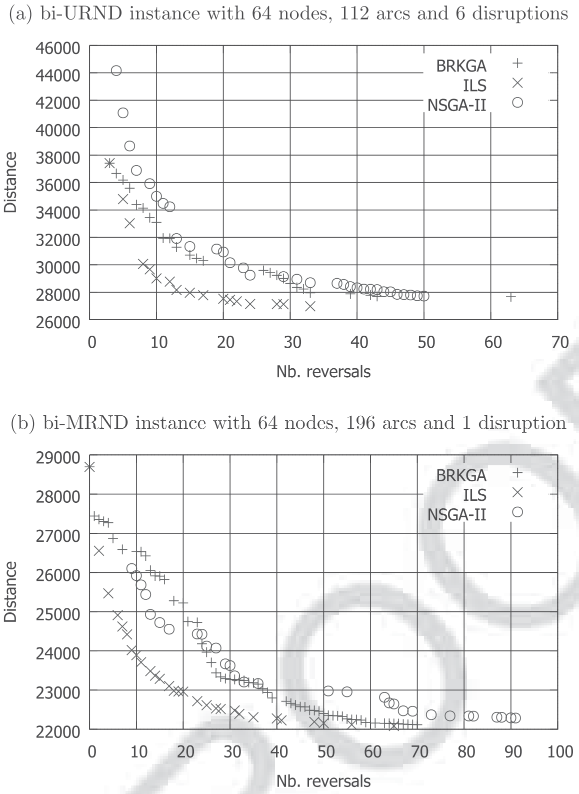
Instance	BKS		NSGA-II				bi-BRKGA				bi-ILS			
	Q	HV	Q	HV	SP	t(s)	Q	HV	SP	t(s)	Q	HV	SP	t(s)
7 × 7-b1-25	22	**0.87**	24	0.73	2.05	**534**	31	0.78	**1.14**	1837	21	0.86	3.82	1725
7 × 7-b1-75	32	**0.90**	37	0.82	1.75	**854**	50	0.84	**0.53**	1276	22	0.89	8.65	3604
7 × 7-b2-25	19	**0.84**	27	0.67	1.59	**535**	26	0.67	1.93	1683	17	**0.84**	**1.40**	1442
7 × 7-b2-75	24	**0.92**	35	0.83	1.47	**861**	46	0.85	**0.55**	1817	21	0.91	2.23	3741
8 × 8-b1-25	15	**0.91**	22	0.68	4.56	**1078**	31	0.71	**3.71**	5029	14	0.90	4.29	4199
8 × 8-b1-75	30	**0.90**	31	0.74	2.86	**1718**	61	0.79	**0.67**	8000	29	**0.90**	2.12	9476
8 × 8-b2-25	19	**0.85**	25	0.71	3.76	**1133**	35	0.67	2.77	3772	16	0.84	**2.63**	3959
8 × 8-b2-75	40	**0.90**	44	0.77	3.36	**1732**	55	0.82	**1.38**	5823	33	**0.90**	9.37	11697

4.3. An application to a real urban network

Experiments have been done on the Troyes instances using the proposed methods. All the four methods: ϵ-constraint, NSGA-II, bi-BRKGA and bi-ILS are used to generate Pareto fronts on troyes-v24 instances, while the larger instances troyes-v58 are tested only by the metaheuristics. Results are reported in Tables 6 and 7.

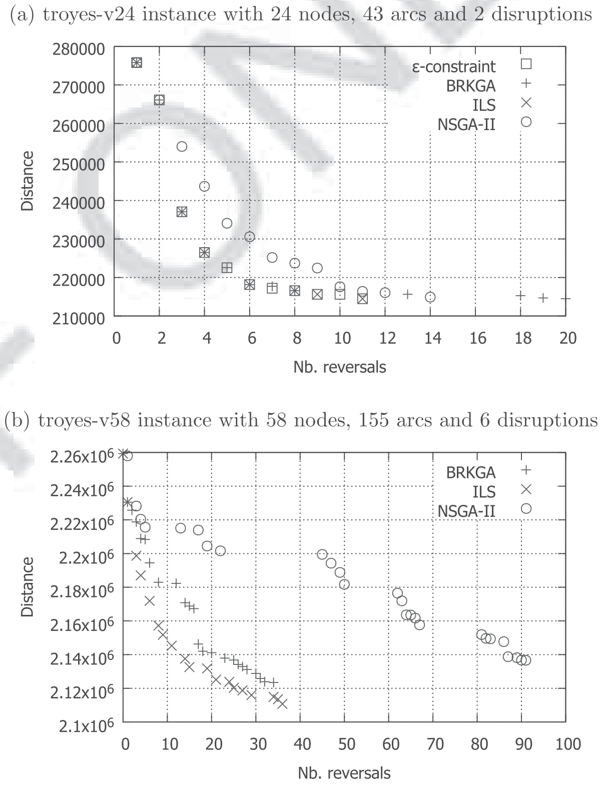
Table 6 reports the results for the four methods on the Troyes instances with 24 nodes. The exact method consumes a relatively larger time compared to the similar instance set of 5×5 multigraphs. bi-BRKGA performs well in terms of hypervolume. However, unlike the behaviour observed on the results for bi-MRND instances, bi-BRKGA does not ensure a good spread of the fronts on these real instances. Bi-ILS got high-quality Pareto fronts in terms of both hypervolume and spacing within a much shorter time than 5×5 bi-MRND instances. The Pareto fronts produced by NSGA-II are generally worse than those from bi-BRKGA and bi-ILS in terms of the hypervolume. Yet, they are better in terms of solutions distribution.

The results for the Troyes instances with 58 nodes are reported in Table 7, BRKGA dominates the other two metaheuristics on the Pareto fronts spread. Bi-ILS still produces good quality fronts but requires more running time. This may not be a drawback since bi-URND and bi-MRND are tactical/strategical problems. It can find Pareto fronts very close to the best-known fronts combining the


(a) bi-URND instance with 64 nodes, 112 arcs and 6 disruptions

(b) bi-MRND instance with 64 nodes, 196 arcs and 1 disruption

Figure 7. Pareto fronts for bi-URND and bi-MRND instances (8x8).


(a) troyes-v24 instance with 24 nodes, 43 arcs and 2 disruptions

(b) troyes-v58 instance with 58 nodes, 155 arcs and 6 disruptions

Figure 8. Pareto fronts for the Troyes instances.

Table 6. Performance comparison of Pareto fronts on troyes-v24 instances.

Instance	ϵ-constraint			NSGA-II				bi-BRKGA				bi-ILS			
	Q	HV	t(s)	Q	HV	SP	t(s)	Q	HV	SP	t(s)	Q	HV	SP	t(s)
troyes-v24-b1	8	**0.86**	108	**8**	**0.86**	**0.36**	73	9	0.85	4.81	82	6	0.85	0.83	**68.05**
troyes-v24-b2	11	**0.86**	263	13	0.78	**0.29**	70	12	0.85	1.49	100	7	0.84	0.79	**61.07**
troyes-v24-b4	7	**0.88**	100	13	0.66	0.89	68	8	0.87	2.21	97	7	0.86	**0.48**	**51.73**
troyes-v24-b6	**9**	**0.86**	291	8	0.84	**0.49**	61	**9**	0.83	2.17	51	8	0.85	0.56	**45.76**

Table 7. Performance comparison of Pareto fronts on troyes-v58 instances.

Instance	BKS		NSGA-II				bi-BRKGA				bi-ILS			
	Q	HV	Q	HV	SP	t(s)	Q	HV	SP	t(s)	Q	HV	SP	t(s)
troyes-v58-b1	17	**0.90**	21	0.50	7.35	**1183**	15	0.77	**1.04**	2230	17	**0.90**	1.10	4754
troyes-v58-b2	16	**0.94**	27	0.61	8.21	**1205**	19	0.84	**1.14**	1553	15	**0.94**	20.29	3413
troyes-v58-b4	**17**	**0.88**	13	0.40	8.08	**1155**	14	0.76	**2.00**	2525	15	**0.88**	2.22	3696
troyes-v58-b6	20	**0.91**	26	0.48	5.18	**1138**	24	0.81	**0.76**	2730	19	**0.91**	1.08	3121

solutions found by all the metaheuristics. The spread of the constructed Pareto fronts are generally reasonable except on the instance with two disruptions. In addition, NSGA-II produces fronts whose hypervolume is significantly lower, compared to bi-BRKGA and bi-ILS, on all the four instances. Same for the spacing, except for one instance where bi-ILS computes a front with a very large spacing, due to a solution far away from the others. Examples of combined Pareto fronts can be found in Figure 8(a) for a troyes-v24 instance and in Figure 8(b) for a troyes-v58 instance. In the latter, we can clearly observe that NSGA-II is inefficient on these instances with the current parameter settings, since there is a relatively large gap between its fronts and the ones built by the other two metaheuristics.

5. Concluding remarks

In this work, the bi-URND and bi-MRND problems are addressed. They correspond respectively to the specific situation in historical centers and in general urban road networks. The total travel distance is considered as the first optimisation criterion. Reversals of arcs direction are introduced to handle the issue of strong connectivity due to disruptions and to help reducing the travel distance. The number of reversals should be limited and it is considered as the second objective.

Based on the bi-objective mathematical model, an exact ϵ-constraint method **is** adapted to construct Pareto-optimal fronts for bi-URND and bi-MRND instances. Moreover, a classical and efficient multi-objective optimisation metaheuristic, NSGA-II, is hence proposed. Two other metaheuristics, BRKGA and ILS, are adapted to address the bi-objective optimisation requirement. Numerical experiments are performed on grid instances with up to 64 nodes and 112 arcs for bi-URND and up to 64 nodes and 224 arcs for bi-MRND. The methods are also applied on instances built from the real road network of Troyes (France). Results show that bi-ILS is able to produce high quality Pareto fronts in both bi-URND and bi-MRND. In addition, it is less dependent on the network configuration, either the topology or the number of disruptions. bi-BRKGA is a solid alternative, even if the fronts it computes are slightly worse. NSGA-II, in its current setting, does not converge well towards the optimal or best Pareto front. It would be relevant to adapt bi-ILS to other problems in order to check if its efficiency still holds.

There is still room for improving the performance of the proposed metaheuristics, which can be addressed in future works. Furthermore, additional realistic constraints and criteria are being studied for the application to general networks, such as arc capacities, multimodal flows and scheduling disruptions, in order to better capture the real context of urban road network problems.

Notes

1. Innov'Action AGIR project (système d'Aide à la décision pour la Gestion d'Interruptions Routière)
2. An open-source Geographical Information System

Disclosure statement

No potential conflict of interest was reported by the author(s).

Acknowledgements

The authors would like to thank the anonymous reviewers for their careful reading of our manuscript and their many insightful comments and suggestions.

References

Ádám, A. (1963). Theory of graphs and its applications. In: *Symp. Smolenice*, Prague, p. 157

Bang-Jensen, J., & Gutin, G. Z. (2008). *Digraphs: Theory, algorithms and applications*, 2nd ed. Springer Publishing Company, Incorporated

Bean, J. C. (1994). Genetic algorithms and random keys for sequencing and optimization. *ORSA Journal on Computing*, *6*(2), 154–160. doi:10.1287/ijoc.6.2.154

Bérubé, J. F., Gendreau, M., & Potvin, J. Y. (2009). An exact ϵ-constraint method for bi-objective combinatorial optimization problems: Application to the traveling salesman problem with profits. *European Journal of Operational Research*, *194*(1), 39–50. doi:10.1016/j.ejor.2007.12.014

Boesch, F., & Tindell, R. (1980). Robbins's theorem for mixed multigraphs. *The American Mathematical Monthly*, *87*(9), 716–719. doi:10.2307/2321858

Chankong, V., & Haimes, Y. Y. (1983). *Multiobjective Decision Making: Theory and Methodology.*: North-Holland: Elsevier Science Publishing.

Chvátal, V., & Thomassen, C. (1978). Distances in orientations of graphs. *Journal of Combinatorial Theory, Series B*, *24*(1), 61–75. doi:10.1016/0095-8956(78)90078-3

Coco, A. A., Duhamel, C., & Santos, A. C. (2019). Modeling and solving the multi-period disruptions scheduling problem on urban networks. *Annals of Operations Research*, doi:10.1007/s10479-019-03248-5

Conte, A., Grossi, R., Marino, A., Rizzi, R., & Versari, L. (2016). Directing road networks by listing strong orientations. In: VeliMäkinen, Puglisi SJ, Salmela L (eds) Combinatorial Algorithms – 27th International Workshop, IWOCA 2016, Helsinki, Finland, August 17–19, 2016, Proceedings, Springer, Lecture Notes in Computer Science (vol. 9843, pp. 83–95).

Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009). *Introduction to algorithms* (3rd ed.). The MIT Press

De Sousa, E. G., Santos, A. C., & Aloise, D. J. (2015). An exact method for solving the bi-objective minimum diameter-cost spanning tree problem. *RAIRO –*

Operations Research, *49*(1), 143–160. doi:10.1051/ro/2014029

Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, *6*(2), 182–197. doi:10.1109/4235.996017

Ehrgott, M., & Gandibleux, X. (2000). A survey and annotated bibliography of multiobjective combinatorial optimization. *OR-Spektrum*, *22*(4), 425–460. doi:10.1007/s002910000046

Gonçalves, J. F., & Resende, M. G. C. (2011). Biased random-key genetic algorithms for combinatorial optimization. *Journal of Heuristics*, *17*(5), 487–525. doi:10.1007/s10732-010-9143-1

Haimes, Y., Lasdon, L., & Wismer, D. (1971). On a bicriterion formulation of the problems of integrated system identification and system optimization. *IEEE Transactions on Systems, Man and Cybernetics SMC-1*, (3), 296–297.

Huang, Y., Santos, A. C., & Duhamel, C. (2016a). *A bi-objective model to address disruptions on unidirectional road networks*. Paper presented at the 8th IFAC Conference on Manufacturing Modelling, Management and Control MIM 2016, Troyes, France, 12, vol. 49, pp. 1620–1625 doi:10.1016/j.ifacol.2016.07.812

Huang, Y., Santos, A. C., & Duhamel, C. (2016b). *Disruptions management in multidirectional road networks*. Proceedings of the 9th Triennial Symposium on Transportation Analysis (TRISTAN), Aruba, Netherlands, p. 4

Huang, Y., Santos, A. C., & Duhamel, C. (2018). Methods for solving road network problems with disruptions. *Electronic Notes in Discrete Mathematics*, *64*, 175–184. 8th International Network Optimization Conference - INOC 2017

Huang, Y., Santos, A. C., & Duhamel, C. (2019). Model and methods to address urban road network problems with disruptions. *International Transactions in Operational Research*, doi:10.1111/itor.12641

Junior, H. V., & Lins, M. P. E. (2009). A win-win approach to multiple objective linear programming problems. *Journal of the Operational Research Society*, *60*(5), 728–733. doi:10.1057/palgrave.jors.2602589

Lourenço, H. R., Martin, O. C., & Stützle, T. (2003). *Iterated local search*. Boston, MA: Springer US, pp. 320–353

Lourenço, H. R., Martin, O. C., & Stützle, T. (2010). *Iterated local search: Framework and applications*. Boston, MA: Springer US, pp 363–397

Mladenović, N., & Hansen, P. (1997). Variable neighborhood search. *Computers & Operations Research*, *24*(11), 1097–1100. doi:10.1016/S0305-0548(97)00031-2

Oliveira, S. L. C., & Ferreira, P. A. V. (2000). Bi-objective optimisation with multiple decision-makers: a convex approach to attain majority solutions. *Journal of the Operational Research Society*, *51*(3), 333–340. doi:10.2307/254091

Pareto, V. (1906). *Manuale di economia politica*. Piccola biblioteca scientifica, Società Editrice Libraria Q7

Resende, M. G. C. (2012). Biased random-key genetic algorithms with applications in telecommunications. *TOP*, *20*(3), 130–153. doi:10.1007/s11750-011-0176-x

Robbins, H. (1939). A theorem on graphs, with an application to a problem on traffic control. *The American Mathematical Monthly*, *46*(5), 281–283. doi:10.2307/2303897

Roberts, F. (1978). Graph theory and its applications to problems of society, CBMS-NSF Regional Conference Series in Applied Mathematics, vol 29. Philadelphia, PA: Society for Industrial and Applied Mathematics (SIAM),.

Santos, A. C., Duhamel, C., & Prins, C. (2013). *Heuristics for setting directions in urban networks*. The X Metaheuristics International Conference (MIC), Singapore, pp 1–4

Santos, A. C., Lima, D. R., & Aloise, D. J. (2014). Modeling and solving the bi-objective minimum diameter-cost spanning tree problem. *Journal of Global Optimization*, *60*(2), 195–216. doi:10.1007/s10898-013-0124-4

Spears, V. M., & Jong, K. A. D. (1991). *On the virtues of parameterized uniform crossover*. Proceedings of the Fourth International Conference on Genetic Algorithms, pp 230–236

Srinivas, N., & Deb, K. (1994). Multiobjective optimization using nondominated sorting in genetic algorithms. *Evol Comput*, *2*(3), 221–248. doi:10.1162/evco.1994.2.3.221

Veldhuizen, D. A. V., & Lamont, G. B. (2000). *On measuring multiobjective evolutionary algorithm performance*. Proceedings of the 2000 Congress on Evolutionary Computation. CEC00 (Cat. No.00TH8512), vol 1, pp 204–211

Zhou, A., Qu, B. Y., Li, H., Zhao, S. Z., Suganthan, P. N., & Zhang, Q. (2011). Multiobjective evolutionary algorithms: A survey of the state of the art. *Swarm and Evolutionary Computation*, *1*(1), 32–49. doi:10.1016/j.swevo.2011.03.001

Zitzler, E., & Thiele, L. (1998). *Multiobjective optimization using evolutionary algorithms — a comparative case study*, Berlin: Springer Berlin Heidelberg, pp. 292–301

Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C. M., & da Fonseca, V. G. (2003). Performance assessment of multiobjective optimizers: An analysis and review. *IEEE Transactions on Evolutionary Computation*, *7*(2), 117–132. doi:10.1109/TEVC.2003.810758

Zopounidis, C., & Pardalos, P. (2010). *Handbook of multicriteria analysis, applied optimization, vol 103*. Berlin: Springer