Available online at www.sciencedirect.com

ScienceDirect

Transportation Research Procedia 47 (2020) 11–18

Transportation Research Procedia

www.elsevier.com/locate/procedia

22nd EURO Working Group on Transportation Meeting, EWGT 2019, 18-20 September 2019, Barcelona, Spain

Computational benchmarking of exact methods for the bilevel discrete network design problem

David Rey[a,b,*]

[a] *School of Civil and Environmental Engineering, UNSW Sydney, NSW 2052, Australia*
[b] *Univ Lyon, UMR-T9401, ENTPE, IFSTTAR, LICIT, F-69675 Lyon, France*

Abstract

The discrete network design problem (DNDP) is a well-studied bilevel optimization problem in transportation. The goal of the DNDP is to identify the optimal set of candidate links (or projects) to be added to the network while accounting for users' reaction as governed by a traffic equilibrium. Several approaches have been proposed to solve the DNDP exactly using single-level, mixed-integer programming reformulations, linear approximations of link travel time functions, relaxations and decompositions. To date, the largest DNDP instances solved to optimality remain of small scale and existing algorithms are no match to solve realistic problem instances involving large numbers of candidate projects. In this work, we examine the literature on exact methodologies for the DNDP and attempt to categorize the main approaches employed. We introduce a new set of benchmarking instances for the DNDP and implement three solution methods to compare computational performance and outline potential directions for improvement. For reproducibility purposes and to promote further research on this challenging bilevel optimization problem, all implementation codes and instance data are provided in a publicly available repository.

Keywords: Network design problem; traffic assignment; bilevel optimization; benchmarking

1. Introduction

We consider the discrete network design problem (DNDP) which is a challenging problem in transportation, introduced by Leblanc (1975). The DNDP can be formulated as a bilevel optimization problem where the leader problem aims to identify the optimal network design to minimize network travel time and the follower problem represents network users' reaction, typically as a static traffic assignment problem (TAP) under user equilibrium (Wardrop, 1952).

The DNDP can be defined on a network with nodes N and directed links A as a multi-commodity network flow problem with nonlinear link travel time functions. Let D be the set of destination nodes and d_{is} be the demand from

* Corresponding author.
E-mail address: d.rey@unsw.edu.au

10.1016/j.trpro.2020.03.067

node $i \in N$ to destination node $s \in D \subseteq N$. If the pair (i, s) is not an Origin-Destination (OD) pair in the network then $d_{is} = 0$ and to ensure flow conservation we set $d_{ss} = -\sum_{i \in N} d_{is}$. We denote $x_{ij,s}$ the flow of on link $(i, j) \in A$ travelling to destination $s \in D$, and x_{ij} the total flow on link $(i, j) \in A$. Let t_{ij} represent the travel time on link $(i, j) \in A$, typically modelled as a strictly convex function of the total link flow x_{ij} to ensure the uniqueness of the equilibrium link flows. Let A_1 be the set of existing links and A_2 be the set of candidate links to improve the network, $A = A_1 \cup A_2$. For each link $(i, j) \in A_2$, let g_{ij} be the cost of adding this link to the network and let $y_{ij} \in \{0, 1\}$ be the variable representing this choice. Let B be the available budget for optimization. In the resulting formulation DNDP, $\mathcal{L}$ is the *leader* problem and $\mathcal{F}$ is the *follower* problem.

The leader problem $\mathcal{L}$ aims to minimize the total network travel time defined as the sum of $x_{ij}t_{ij}(x_{ij})$ over all links, subject to a budget constraint capturing the cost of link addition decisions $\boldsymbol{y}$—hereby referred to as the leader variable. The link flow pattern variable $\boldsymbol{x} = [x_{ij}]_{(i,j) \in A}$ is optimized in the follower problem $\mathcal{F}$, which is the traditional link-based TAP formulation under UE (Beckmann et al., 1956; Leblanc, 1975; Magnanti and Wong, 1984). The impact of the leader variable $\boldsymbol{y}$ in the follower is achieved through the constraint $x_{ij} \leq y_{ij}M$ wherein M is an upper-bound on the total link flow x_{ij}.

$$
\begin{aligned}
(\mathcal{L})\ \min_{\boldsymbol{y}} \quad & \sum_{(i,j)\in A} x_{ij} t_{ij}(x_{ij}) && \\
\text{s.t.} \quad & \sum_{(i,j)\in A_2} y_{ij} g_{ij} \leq B && \\
& y_{ij} \in \{0,1\} && \forall (i,j) \in A_2 \\
(\mathcal{F}) \qquad & \boldsymbol{x} \in \arg\min_{\boldsymbol{x}} \ \sum_{(i,j)\in A} \int_0^{x_{ij}} t_{ij}(v)dv && \\
& \text{s.t.} \quad \sum_{j \in N:(i,j)\in A} x_{ij,s} - \sum_{j \in N:(j,i)\in A} x_{ji,s} = d_{is} && \forall i \in N, \forall s \in D \\
& \qquad \sum_{s\in D} x_{ij,s} = x_{ij} && \forall (i,j) \in A \\
& \qquad x_{ij} \leq y_{ij} M && \forall (i,j) \in A_2 \\
& \qquad x_{ij,s} \geq 0 && \forall (i,j) \in A, \forall s \in D
\end{aligned}
\tag{DNDP}
$$

The goal of this paper is threefold. The first objective is to present a summary of the state-of-the-art of exact methodologies for the DNDP (Section 2). The second objective is to discuss existing methodologies, including mathematical formulations and solution algorithms, and propose a new taxonomy of exact methods for the DNDP (Section 3). The third objective is to conduct a computational benchmark of selected methodologies to highlight existing computational bottlenecks and outline future research directions. To promote knowledge exchange and ensure results reproducibility, all codes of implemented methodologies, as well as new benchmark instances for the DNDP, are provided in a public repository available at `https://github.com/davidrey123/DNDP`. In achieving these objectives, this paper aims to provide a new foundations to engage the scientific community and to foster research in developing novel exact and scalable methodologies for the DNDP.

2. State-of-the-art

Several efforts have been proposed to solve the DNDP to global optimality. The seminal work of Leblanc (1975) introduced a Branch-and-Bound (B&B) algorithm for the DNDP which used the system-optimum (SO) relaxation of the TAP to find lower bounds. This relaxation requires fixing all unfixed $\boldsymbol{y}$ variables to 1 to avoid Braess' paradox effects (Braess, 1968), and may thus lead to poor lower bounds. Gao et al. (2005) introduced a mixed-integer nonlinear programming (MINLP) approach based on generalized Benders' decomposition. The authors proposed a benchmark network with a single origin-destination (OD) pair, 12 nodes, 17 existing links and 6 candidate links, commonly referred to as *Gao's instance*. They also provided results for an instance based on the Sioux Falls network which contains 24 zones and nodes, 76 existing links and 5 candidate projects (which may involve one or more links) but computation time is not reported. It was noted by Farvaresh and Sepehri (2013) that despite claims of global optimality, the proposed approach of Gao et al. (2005) may converge to local optimums.

Bilevel optimization problems with a convex follower problem can be reformulated into single-level formulations by representing the follower problem using its Karush-Kuhn-Tucker (KKT) conditions and introducing binary variables to model complementarity slackness conditions (Bard, 2013). This applies to the DNDP since the follower problem is a TAP which can be represented as a convex nonlinear program (NLP) (Beckmann et al., 1956). Farvaresh and Sepehri (2011) proposed a mixed-integer linear programming (MILP), single-level reformulation of the DNDP obtained using piecewise linear approximations of link travel time functions. Numerical results are only reported for Gao's instance and an extended network based on Gao's instance which contains 16 nodes, 17 existing links and 25 candidate links, and 2 OD pairs. Farvaresh and Sepehri (2013) proposed a more scalable approach which extends the seminal B&B of Leblanc (1975) by solving the SO relaxation of the DNDP—instead of the TAP—as a mixed-integer nonlinear program (MINLP) to provide tighter lower bounds. This extended B&B algorithm was shown capable to solve instances with up to 100 nodes, 387 links and 15 candidate projects.

Luathep et al. (2011) proposed an SO-relaxation based approach wherein variational inequalities (VIs) are iteratively added to ensure UE conditions. Numerical results are reported for Gao's instance, as well as for an instance on Sioux Falls network with 5 candidate projects, and a variation with 10 candidate projects for a mixed (discrete and continuous) case. Fontaine and Minner (2014) proposed a Benders' decomposition of a single-level MILP reformulation of a linearized DNDP with piecewise linear approximations of link travel time functions. Using this decomposition, the authors are able to solve this linearized DNDP on a network of Berlin's centre containing 36 zones, 398 nodes, 871 links and 10 candidate projects.

Wang et al. (2013) studied an extended DNDP where the capacity of candidate links must also be decided. They present two global optimization algorithms which are based on the SO-relaxation of the DNDP, as well as a dynamic outer approximation of link travel time functions to derive lower bounds. Numerical results for a Sioux Falls network instance with up to 10 candidates links and 3 levels of capacity are reported. Wang et al. (2015) extended the DNDP to a variant involving both discrete and continuous decisions variables for adding links and determining their capacity, respectively. The authors used the VI formulation of Luathep et al. (2011) but combine it with an outer approximation of link travel time functions. Numerical results are only reported for Gao's instance. Bagloee et al. (2017) proposed a B&B algorithm which uses a generalized Benders' decomposition approach to solve the SO-relaxation of the DNDP at each node of the tree. Results on Sioux Falls and Winninpeg's network are reported with up to 20 candidate projects.

Although several efforts have been proposed to solve the DNDP or a linear approximation of this problem, the literature on exact or near-exact approaches remains scarce. Further, there is no reference datasets for benchmarking solution methods which undermines the research on this challenging bilevel optimization problem.

3. Exact methodologies for the DNDP

In this section, we discuss formulations and algorithms to the solve the DNDP to optimality. We start by discussing the role of link travel time functions and attempt to categorize existing solution methodologies for the DNDP.

3.1. Link travel time functions approximations

A major computational challenge inherent to the DNDP is the nonlinearity of the link travel time functions $t_{ij}(x_{ij})$, typically Bureau of Public Roads (BPR) functions. The consensus on link travel time functions is to use strictly convex functions of the form $t_{ij}(x_{ij}) = T_{ij} + c_{ij}x_{ij}^{e_{ij}}$ where T_{ij} represents link free-flow travel time, c_{ij} is a coefficient which captures link capacity and $e_{ij} \geq 1$ is an exponent; to ensure uniqueness of the link flow solution. The vast majority of existing data for link travel time functions of this form assumes $e_{ij} = 4$ for all $(i, j) \in A$. Link travel time functions are present in the objective function of both leader and follower problems in the form $x_{ij}t_{ij}(x_{ij})$ which involves the nonlinear term $x_{ij}^{e_{ij}+1}$. This term, and sometimes term $x_{ij}^{e_{ij}}$, have been either approximated using piecewise linear functions (Farvaresh and Sepehri, 2011; Luathep et al., 2011; Fontaine and Minner, 2014) or using linear outer approximation schemes (Wang et al., 2013, 2015).

Piecewise linear approximations require discretizing the domain of link flow variable $\boldsymbol{x}$ into m disjunctive segments and requires $O(m)$ auxiliary variables to adjust the approximated nonlinear term according to the segment activated. The domain of link flows is typically defined as $x_{ij} \in [0, \bar{x}_{ij}]$ where $\bar{x}_{ij}$ is the maximum flow that can travel on link (i, j). Determining tight values for the upper bound $\bar{x}_{ij}$ is not trivial and using a conservative value such as the total demand of the network may lead to poor approximation schemes. However, since the nonlinear terms $x_{ij}^{e_{ij}}$ and $x_{ij}^{e_{ij}+1}$ are convex on $[0, \bar{x}_{ij}]$, piecewise linear approximations of these terms do not require any integer variables, as noted by Farvaresh and Sepehri (2011) and in Fontaine and Minner (2014). Hence, while piecewise linear approximation may require a significant amount of additional variables and linear constraints to achieve high-quality solutions, their impact on computational performance can be moderate.

Outer approximations of link travel time functions attempt to derive compact convex envelopes of nonlinear terms $x_{ij}^{e_{ij}}$ and $x_{ij}^{e_{ij}+1}$ using integer-linear constraints. These schemes are notoriously computationally challenging but guarantee that the original (non-linearized) DNDP is solved to optimality. Only in rare cases have link travel time functions been incorporated without any direct approximation scheme (Farvaresh and Sepehri, 2013; Bagloee et al., 2017).

We next discuss exact methodologies for the DNDP which have adopted one of the above approaches to handle link travel time functions. Since piecewise linear approximations of link travel time functions cannot be guaranteed to converge to optimal solutions unless $m \to \infty$, such methodologies are referred to as linearized DNDP approaches.

3.2. SO-relaxation based approaches

Despite its potential weak initial lower bound, the SO-relaxation of the DNDP has emerged as a powerful mechanism to conceive iterative solution methods for the DNDP. The SO-relaxation of the DNDP is a single-level optimization problem which ignores the follower objective function as summarized in SO-DNDP.

$$\begin{aligned}
\min_{\boldsymbol{x},\boldsymbol{y}} \quad & \sum_{(i,j)\in A} x_{ij} t_{ij}(x_{ij}) && \\
\text{s.t.} \quad & \sum_{(i,j)\in A_2} y_{ij} g_{ij} \leq B && \\
& \sum_{j\in N:(i,j)\in A} x_{ij,s} - \sum_{j\in N:(j,i)\in A} x_{ji,s} = d_{is} && \forall i \in N, \forall s \in D \\
& \sum_{s\in D} x_{ij,s} = x_{ij} && \forall (i,j) \in A \\
& x_{ij} \leq y_{ij} M && \forall (i,j) \in A_2 \\
& y_{ij} \in \{0,1\} && \forall (i,j) \in A_2 \\
& x_{ij,s} \geq 0 && \forall (i,j) \in A, \forall s \in D
\end{aligned} \tag{SO-DNDP}$$

3.2.1. Branch and bound

Solving SO-DNDP yields a lower bound on the optimum of DNDP which can serve as starting point of iterative schemes. Leblanc (1975) was the first to propose a customized B&B algorithm which branches on unfixed $\boldsymbol{y}$ variables to yield subproblems that refine the initial lower bound. At each node of the B&B tree, Leblanc (1975) approach further relaxed SO-DNDP by temporarily fixing all unfixed $\boldsymbol{y}$ variables and solving the resulting SO-TAP using a convex programming algorithm, e.g. Frank-Wolfe. This B&B scheme was refined and extended by Farvaresh and Sepehri (2013) which proposed to solve SO-DNDP at each node of the tree using a global MINLP algorithm. Upper bounds are obtained by solving the TAP under UE conditions after obtaining $\boldsymbol{y}$ from SO-DNDP.

Bagloee et al. (2017) proposes a B&B algorithm which uses generalized Benders' decomposition approach to solve the SO-relaxation of the DNDP at each node of the tree. The proposed algorithm is parameterized using a value (denoted α) which influences the solution of the TAP solved therein. The algorithm is not guaranteed to find global optimal solutions if $\alpha \neq 1$.

3.2.2. Variational inequalities

Luathep et al. (2011) proposed a cutting-plane algorithm to complement formulation SO-DNDP with VIs that characterizes UE conditions (Dafermos, 1980). Let $\Omega_{\mathcal{F}}$ be the polyhedron of the feasible region of the follower problem $\mathcal{F}$ and let $\boldsymbol{x}^\star$ be the UE link flow pattern, the following VI holds.

$$\sum_{(i,j)\in A} t_{ij}(x^\star_{ij})(x_{ij} - x^\star_{ij}) \geq 0 \qquad \forall \boldsymbol{x} \in \Omega_{\mathcal{F}} \tag{1}$$

The approach of Luathep et al. (2011) is based on the observation that VIs corresponding to the set of extreme points of the TAP polyhedron is sufficient and necessary to characterize UE conditions. At each iteration, SO-DNDP is solved with a restricted set of VIs and shortest path problems are solved for each OD pair to identify violated VIs of the form (1), which are then added as cuts to SO-DNDP until none can be found.

3.2.3. Interdiction cuts

Wang et al. (2013) observed that it was sufficient to iteratively forbid the last $\boldsymbol{y}$ solution found to construct a UE solution from the SO-relaxation of the DNDP. At each iteration, the proposed algorithm first solves SO-DNDP before calculating the UE cost of the corresponding $\boldsymbol{y}$ solution using any TAP algorithm to obtain an upper bound. The process is then repeated by adding an interdiction constraint to identify the next-best SO network design. Let $\boldsymbol{y}^k$ be the optimal solution of SO-DNDP at iteration k. At iteration $k+1$, SO-DNDP is solved with the interdiction constraint

$$\sum_{(i,j)\in A_2} \left(y_{ij}(1 - y^n_{ij}) + (1 - y_{ij})y^n_{ij} \right) \geq 1 \qquad \forall n \in \{1, \ldots, k\} \tag{2}$$

The iterative process is repeated until the resulting objective value is greater than or equal to the upper bound.

3.3. KKT conditions based approaches

Since the TAP can be formulated as a convex optimization problem, an intuitive approach to solve the DNDP is to replace the follower problem $\mathcal{F}$ by its KKT conditions. Farvaresh and Sepehri (2011) proposed such a direct approach wherein the KKT conditions of the TAP are represented with auxiliary variables. Let $\pi_{is} \geq 0$ be the travel time or path travel time from node $i \in N$ to destination $s \in D$. The UE conditions of the TAP require:

$$t_{ij}(x_{ij}) - \pi_{is} + \pi_{js} \geq 0 \qquad \forall (i,j) \in A, \forall s \in D \tag{3a}$$

$$x_{ij}(t_{ij}(x_{ij}) - \pi_{is} + \pi_{js}) = 0 \qquad \forall (i,j) \in A, \forall s \in D \tag{3b}$$

The complementarity slackness conditions (3b) is nonlinear and typical mathematical programming approaches require additional binary variables to obtain an integer-linear form suitable for MILP. Farvaresh and Sepehri (2011) observe that this surplus of binary variables yields considerable computational challenge. Fontaine and Minner (2014) proposed an alternative approach to incorporate the KKT conditions of the TAP which consists of replacing the objective of the follower problem $\mathcal{F}$ with a primal-dual constraint which requires a null duality gap. This method is applied to a linearized DNDP, which results in a single-level MILP composed of leader, primal follower and dual follower constraints and variables, with the addition of the primal-dual constraint. This MILP does not require any additional binary variable and the authors propose a Benders' decomposition approach.

A summary of the solution methods discussed is provided in Table 1. Most of these methodologies can be adapted to work with piecewise linear approximations or outer-approximations of link travel time functions. While the former only solves the so-called linearized DNDP, piecewise linear approximations are often capable of identifying global optimal solutions of the original problem and are, at the same time, substantially easier to implement. In the remaining of this paper, we focus on the linearized DNDP and present some computational results on new instances of the DNDP.

Paper	Method	Link travel time function approximation	Type(s) of optimization problem solved
Leblanc (1975)	SO-relaxation and B&B	N/A	Convex NLP
Gao et al. (2005)	Generalized Benders' decomposition	N/A	MINLP and Convex NLP
Farvaresh and Sepehri (2011)	KKT conditions	Piecewise linear	MILP
Luathep et al. (2011)	SO-relaxation and VIs	Piecewise linear	MILP and LP
Farvaresh and Sepehri (2013)	SO-relaxation and B&B	N/A	MINLP and Convex NLP
Wang et al. (2013)	SO-relaxation and interdiction cuts	Outer linear	MILP and Convex NLP
Fontaine and Minner (2014)	KKT conditions and Benders' decomposition	Piecewise linear	MILP and LP
Wang et al. (2015)	SO-relaxation and VIs	Outer linear	MILP and LP
Bagloee et al. (2017)	B&B and Generalized Benders' decomposition	N/A	MINLP and Convex NLP

Table 1. Summary of exact methodologies for the DNDP.

4. Computational benchmarking of the linearized DNDP

4.1. Algorithms implementation

We implement three solution methods for the linearized DNDP: i) the B&B algorithm of Farvaresh and Sepehri (2013) which is an extension of Leblanc (1975)—SOBB; ii) the SO-relaxation based algorithm of Wang et al. (2013) with interdiction cuts—SOIC, and iii) the primal-dual formulation of Fontaine and Minner (2014) as a single-level MILP (without Benders' decomposition)—MKKT. All three solution methods, SOBB, SOIC and MKKT, are implemented using the piecewise linear approximation of link travel time functions proposed by Farvaresh and Sepehri (2011). Hence, in all cases, the optimization problems solved are MILPs and LPs (TAPs within SOBB and SOIC are solved in their linearized form). This differs from the original implementation of Farvaresh and Sepehri (2013) and Wang et al. (2013), wherein outer approximation schemes are used. In addition, the method MKKT is implemented as direct MILP approach unlike the Benders' decomposition scheme proposed in Fontaine and Minner (2014). To measure the quality of the approximated solutions, the flow pattern corresponding to the best (lowest leader objective value) $\boldsymbol{y}$ solution among all three methods is calculated by solving the TAP as a convex problem.

All methods are implemented in Python. All MILPs and LPs are solved using CPLEX 12.8. Convex TAPs are solved using the Pyomo module and IPOPT. All solution methods were tested and implemented on the same Windows 7 machine with 16Gb of RAM and a CPU of 2.7Ghz, in a single-thread mode with a time limit of 10 minutes. The upper bound on link flows $\bar{x}_{ij}$ was set to $1e^5$ and this value is also used for M. The number of segments used in the piecewise linear approximations of link travel time functions is $m = 100$. A scaling factor of $1e^{-3}$ is used to scale travel demand and link capacities as it was found to improve computational performance. For reproducibility purposes, all implemented optimization formulations and codes are provided at `https://github.com/davidrey123/DNDP`.

4.2. Benchmark instances

New instances for the DNDP based on Sioux Falls network[1] have been designed to test the implemented solution methods. For these new instances, a total of 15 pairs of new links (total of 30 links) have been created, along with their performance characteristics and addition costs. Among these new links, 5 of the 15 pairs are identical to the ones used by Luathep et al. (2011). A total of 20 DNDP instances have been created, 10 of these contain 10 new links (5 pairs) and are named `SF_DNDP_10`, and the remaining contain 20 new links (10 pairs) and are named `SF_DNDP_20`. In all our numerical experiments, link addition variables are not paired, i.e. the size of A_2 is equal to the total number of new links which may or may not be added in pairs; hence instances `SF_DNDP_10` require 10 binary variables and

[1] These instances are based on Sioux Falls network as available at `https://github.com/bstabler/TransportationNetworks`

	SF_DNDP_10									SF_DNDP_20								
	$B_\%$ = 25%			$B_\%$ = 50%			$B_\%$ = 75%			$B_\%$ = 25%			$B_\%$ = 50%			$B_\%$ = 75%		
Method	AT	AG	TO	AT	AG	TO	AT	AG	TO	AT	AG	TO	AT	AG	TO	AT	AG	TO
SOBB	56.8	0.0	0.0	333.8	0.39	20.0	484.9	1.75	60.0	600.0	2.48	100.0	600.0	4.76	100.0	600.0	6.07	100.0
SOIC	33.8	0.0	0.0	283.4	0.31	20.0	357.5	0.69	40.0	600.0	3.76	100.0	600.0	4.44	100.0	600.0	5.26	100.0
MKKT	57.2	0.0	0.0	170.6	0.0	0.0	155.8	0.0	0.0	600.0	5.68	100.0	600.0	6.46	100.0	600.0	6.21	100.0

Table 2. Budget sensitivity experiment. Methods SOBB, SOIC and MKKT are implemented for instances SF_DNDP_10 and SF_DNDP_20 for a budget B equal to $B_\%$ = 25%, 50% and 75% of the total cost $\sum_{(i,j)\in A_2} g_{ij}$. The time limit is 10 minutes. AT is the average runtime in seconds, AG is the average relative optimality gap upon termination in % and TO is the proportion of time-outs in %.

	SF_DNDP_10									SF_DNDP_20								
	$D_\%$ = 50%			$D_\%$ = 100%			$D_\%$ = 150%			$D_\%$ = 50%			$D_\%$ = 100%			$D_\%$ = 150%		
Method	AT	AG	TO	AT	AG	TO	AT	AG	TO	AT	AG	TO	AT	AG	TO	AT	AG	TO
SOBB	537.7	0.54	80.0	336.6	0.40	20.0	79.2	0.0	0.0	600.0	2.38	100.0	600.0	4.75	100.0	432.2	0.24	40.0
SOIC	513.8	0.56	80.0	284.6	0.34	20.0	14.6	0.0	0.0	600.0	2.14	100.0	600.0	4.44	100.0	254.9	0.12	30.0
MKKT	87.9	0.0	0.0	172.1	0.0	0.0	65.9	0.0	0.0	600.0	2.58	100.0	600.0	6.46	100.0	519.8	1.43	50.0

Table 3. Demand sensitivity experiment. Methods SOBB, SOIC and MKKT are implemented for instances SF_DNDP_10 and SF_DNDP_20 for a budget B equal to 50% of the total cost $\sum_{(i,j)\in A_2} g_{ij}$ and a demand of $D_\%$ = 50%, 100% and 150% of the base demand. The time limit is 10 minutes. AT is the average runtime in seconds, AG is the average relative optimality gap upon termination in % and TO is the proportion of time-outs in %.

instances SF_DNDP_20 require 20 binary variables. All instance data is provided in the public repository available at https://github.com/davidrey123/DNDP in the TNTP format. An extra column has been appended to provide link cost data (g_{ij}) and a value of 0 indicates that the link is part of the original network (before optimization).

4.3. Numerical results

Two numerical experiments are conducted on instances SF_DNDP_10 and SF_DNDP_20: a budget sensitivity analysis wherein the available budget B is equal to 25%, 50% and 75% of the total cost $\sum_{(i,j)\in A_2} g_{ij}$, and a demand sensitivity analysis wherein the budget is fixed to 50% of the total cost and travel demand is set to 50%, 100% and 150% of the base demand. The average performance of all three methods implemented, i.e. SOBB, SOIC and MKKT, is reported in Tables 2 and 3 for budget and demand sensitivity analysis, respectively.

We find that 10-project instances SF_DNDP_10 can present considerable computational challenges, especially for SO-relaxation based methods when the budget available is relatively high ($B_\% = 75\%$). Increasing the number of candidate links to 20 results in substantial difficulties in most cases, regardless of the method used, since most of the time the methods implemented were not able to converge within the 10-minute time limit. The results of the budget sensitivity analysis on SF_DNDP_10 instances show that MKKT is often faster than other methods at medium and high budgets ($B_\% = 50\%, 75\%$). In turn, SOIC and SOBB tend to time-out more frequently at these budgets, which is likely due to the more substantial gap between UE and SO flow patterns when a high budget is available. At low budget ($B_\% = 25\%$), SOIC outperforms other methods. Instances SF_DNDP_20 result systematically in time-outs and the average optimality gap upon termination tends to increase with the budget. The performance of all three methods is of the same order of magnitude, with optimality gaps in the range of 3%-6%. The demand sensitivity analysis highlights the good performance of MKKT at low demand ($D_\% = 50\%$) compared to other methods on SF_DNDP_10 instances. Increasing the demand to 150% of the base demand yields substantially easier problems, notably for SF_DNDP_20 instances, and SOIC is found to outperfom other methods in terms of number of time-outs.

To assess the quality of the linear approximation of link travel time functions, we compare the network travel time obtained by solving the TAP in its convex NLP form for the best $\boldsymbol{y}$ solution found among all three methods. The deviation observed is on average of 0.28% with a standard deviation of 0.25%. Detailed numerical results are provided in the supplementary material available at https://github.com/davidrey123/DNDP.

5. Discussion and research directions

The DNDP is a challenging bilevel optimization problem with critical implications for network design in urban transport systems. Over the past decade, exact methods have become more capable to solve non-trivial decisions problems. We synthesized the literature on exact methodologies and discussed the main approaches developed to solve the DNDP to optimality. In total, nine papers have been examined. Although some of these papers have not addressed the DNDP in its present form, they have presented a methodology for a related problem which can be reduced to the DNDP. The characteristics of the solution methods examined have been categorized in two types of approaches: SO-relaxation based and KKT conditions based methods. A computational benchmarking of the DNDP was conducted by adapting three solution methods and comparing their performance using the same piecewise linear link travel time function approximation. For this benchmark, a total of 20 new instances for the DNDP have been created. For reproducibility purposes, all implementation formulations, codes and benchmarking instances are available at `https://github.com/davidrey123/DNDP`.

Overall, this study shows that even medium-size instances for the linearized DNDP can present considerable computational challenges. The numerical experiments revealed that some 10-project instances and most 20-project instances could not be solved within the imposed time limit (10 minutes). Although this time limit is relatively small, all methods were implemented using a piecewise linear approximation of link travel time functions which is expected to outperform outer approximations approaches.

Further research is critically needed to develop scalable exact methodologies for the DNDP and its variants. There is ample empirical evidence suggesting that for large enough values of m, the $\boldsymbol{y}$-variable solution of the linearized DNDP is identical to that of the original DNDP (Luathep et al., 2011; Farvaresh and Sepehri, 2011; Fontaine and Minner, 2014); although linearized link travel time functions cannot guarantee unique link flow solutions. Hence, identifying balanced configurations (upper bounds, discretisation scheme) between solution quality and performance a priori to refine the approximation scheme is not trivial and merits further efforts. In addition, while efficient algorithms for the TAP have been proposed, the potential of such algorithms has typically not been integrated in solution methods for the DNDP. Recent work by Rey et al. (2019) has outlined promising avenues to achieve this integration but additional efforts are needed to provide global optimality guarantees on high-dimensional problems.

References

Bagloee, S.A., Sarvi, M., Patriksson, M., 2017. A hybrid branch-and-bound and benders decomposition algorithm for the network design problem. Computer-Aided Civil and Infrastructure Engineering 32, 319–343.

Bard, J.F., 2013. Practical bilevel optimization: algorithms and applications. volume 30. Springer Science & Business Media.

Beckmann, M., McGuire, C.B., Winsten, C.B., 1956. Studies in the Economics of Transportation. Technical Report.

Braess, D., 1968. Über ein paradoxon aus der verkehrsplanung. Unternehmensforschung 12, 258–268.

Dafermos, S., 1980. Traffic equilibrium and variational inequalities. Transportation science 14, 42–54.

Farvaresh, H., Sepehri, M.M., 2011. A single-level mixed integer linear formulation for a bi-level discrete network design problem. Transportation Research Part E: Logistics and Transportation Review 47, 623–640.

Farvaresh, H., Sepehri, M.M., 2013. A branch and bound algorithm for bi-level discrete network design problem. Networks and Spatial Economics 13, 67–106.

Fontaine, P., Minner, S., 2014. Benders decomposition for discrete–continuous linear bilevel problems with application to traffic network design. Transportation Research Part B: Methodological 70, 163–172.

Gao, Z., Wu, J., Sun, H., 2005. Solution algorithm for the bi-level discrete network design problem. Transportation Research Part B: Methodological 39, 479–495.

Leblanc, L.J., 1975. An algorithm for the discrete network design problem. Transportation Science 9, 183–199.

Luathep, P., Sumalee, A., Lam, W.H., Li, Z.C., Lo, H.K., 2011. Global optimization method for mixed transportation network design problem: a mixed-integer linear programming approach. Transportation Research Part B: Methodological 45, 808–827.

Magnanti, T.L., Wong, R.T., 1984. Network design and transportation planning: Models and algorithms. Transportation science 18, 1–55.

Rey, D., Bar-Gera, H., Dixit, V.V., Waller, S.T., 2019. A branch-and-price algorithm for the bilevel network maintenance scheduling problem. Transportation Science, 53, 1455–1478.

Wang, D.Z., Liu, H., Szeto, W., 2015. A novel discrete network design problem formulation and its global optimization solution algorithm. Transportation Research Part E: Logistics and Transportation Review 79, 213–230.

Wang, S., Meng, Q., Yang, H., 2013. Global optimization methods for the discrete network design problem. Transportation Research Part B: Methodological 50, 42–60.

Wardrop, J.G., 1952. Some theoretical aspects of road traffic research, in: Inst Civil Engineers Proc London/UK/.