Transportation Research Part B 70 (2014) 163–172

Contents lists available at ScienceDirect

Transportation Research Part B

journal homepage: www.elsevier.com/locate/trb

Benders Decomposition for Discrete–Continuous Linear Bilevel Problems with application to traffic network design

Pirmin Fontaine *, Stefan Minner

Logistics & SCM, TUM School of Management, Arcisstraße 21, 80333 Munich, Germany

ARTICLE INFO

Article history:
Received 19 December 2013
Received in revised form 9 September 2014
Accepted 10 September 2014

Keywords:
Bilevel Programming
Discrete Network Design Problem
Benders Decomposition
Linear approximation

ABSTRACT

We propose a new fast solution method for linear Bilevel Problems with binary leader and continuous follower variables under the partial cooperation assumption. We reformulate the Bilevel Problem into a single-level problem by using the Karush–Kuhn–Tucker conditions. This non-linear model can be linearized because of the special structure achieved by the binary leader decision variables and subsequently solved by a Benders Decomposition Algorithm to global optimality. We illustrate the capability of the approach on the Discrete Network Design Problem which adds arcs to an existing road network at the leader stage and anticipates the traffic equilibrium for the follower stage. Because of the non-linear objective functions of this problem, we use a linearization method for increasing, convex and non-linear functions based on continuous variables. Numerical tests show that this algorithm can solve even large instances of Bilevel Problems.

1. Introduction

As we are facing increasing population in cities, the demand for transportation increases. This leads to more congested roads and longer travel times. Moreover, congestions lead to air pollution, noise pollution and a lower quality of living. Therefore, traffic networks have to be expanded and an efficient usage of the budget in network expansions should be achieved. In the literature, these problems are addressed as Bilevel Problems (i.e., LeBlanc, 1975; Farahani et al., 2013; Gao et al., 2005; Poorzahedy and Turnquist, 1982; Luathep et al., 2011).

Bilevel Problems are mathematical programming problems consisting of an optimization problem (the upper one – also called the leader) with nested optimization problems (the lower ones – also called follower) in the constraints. In practice, these problems can occur if decentralized or hierarchical decisions have to be taken (Schneeweiß, 2003). First, the leader has to decide over a subset of the decision variables, which affects the feasible region of the follower. Afterwards, the follower has to decide over the other subset of decision variables, which affects the objective value of the leader. Ben-Ayed and Blair (1990) showed that even Linear Bilevel Problems with continuous leader and follower decision variables are $\mathcal{NP}$-hard. In this paper, we consider the linear case with binary leader variables and continuous follower variables (DCBLP). Over the past decades, several methods have been proposed to solve Linear Bilevel Problems (BP). A survey on different solution methods and applications on Bilevel Problems can be found in Colson et al. (2005). Bard and Moore (1990) presented a branch-and-bound approach based on the Kuhn-Tucker conditions of the follower problem and Hansen et al. (1992) proposed a branch-and-bound algorithm which branches on binding follower constraints. Both algorithms were introduced for the continuous BP, but can also be applied for BP with integer leader variables. Recently, Saharidis and Ierapetritou (2009) suggested a

* Corresponding author.
E-mail address: pirmin.fontaine@tum.de (P. Fontaine).

http://dx.doi.org/10.1016/j.trb.2014.09.007
0191-2615/

Benders Decomposition (BD) approach for solving the mixed-integer BP with discrete leader variables and continuous follower variables. However, this approach still has to solve a mixed-integer program (MIP) in the Master Problem and in the Slave Problem, because it solves a Bilevel Problem in the Slave Problem by applying the Active Constraint Strategy (Grossmann and Floudas, 1987). In this paper, we introduce a new solution method for the BP with binary leader variables, which is also using BD, but only solves continuous Slave Problems.

One application of Bilevel Programming are Urban Network Design Problems. These problems improve an existing traffic network by adding new links to the network or increasing the capacity of already existing links. The latter problem is called the Continuous Network Design Problem (CNDP) (Abdulaal and LeBlanc, 1979) and the former, which we address in this paper, the Discrete Network Design Problem (DNDP) (LeBlanc, 1975). All these problems have to be modeled as Bilevel Problems, as the objective of the network designer – to reduce congestion in the network – and the objective of the follower – to find the fastest way from origin to destination – do not have to be the same (Braess et al., 2005). Farahani et al. (2013) recently summarized the different existing models and solution methods.

Abdulaal and LeBlanc (1979) proposed a direct search method for the CNDP and Suh and Kim (1992) presented a descent algorithm for non-linear Bilevel Problems. Moreover, Simulated Annealing and Genetic Algorithms were used to solve the CNDP (Meng and Yang, 2002; Xu et al., 2009; Mathew and Sharma, 2009).

For the DNDP, which is more difficult to solve than the CNDP, (LeBlanc, 1975) proposed a branch-and-bound method. A Generalized Benders Decomposition approach with the use of support functions was proposed by Gao et al. (2005).

Luathep et al. (2011) presented a mixed-integer linear formulation for the DNDP which formulates the follower problem with the variational inequality problem. This formulation approximates the travel time function with piecewise linear terms and relaxes bilinear terms by introducing binary auxiliary variables. Farvaresh and Sepehri (2011) applied the Karush–Kuhn–Tucker conditions to transform the problem into a non-linear MIP. This formulation was further linearized by introducing binary auxiliary variables. Recently, Wang et al. (2013) presented a global optimization approach which uses the relation between the system optimum and the user optimum. Further, Ekström et al. (2012) proposed a global optimization mixed-integer program for a similar problem, the Toll Location Problem.

Since real-size instances are still difficult to solve, approximation algorithms and several heuristics were applied. Poorzahedy and Turnquist (1982) proposed branch-and-bound based heuristic, Poorzahedy and Abulghasemi (2005) applied the Ant Colony System metaheuristic and Poorzahedy and Rouhani (2007) showed that hybrid metaheuristics based on tabu search, simulated annealing and genetic algorithm perform even better on the DNDP.

Besides the contribution to solve considerably larger instances of DCBLPs, we present a formulation of the Discrete Network Design Problem which approximates the non-linear convex objective functions only by piecewise linear terms (without additional binary variables) and can be solved by our algorithm. Compared to Luathep et al. (2011) and Farvaresh and Sepehri (2011), we avoid introducing binary auxiliary variables and the relaxation of bilinear terms. Because of the very small number of binary variables, the linear MIP formulation of the DNDP has computational benefits and we can solve even large instances for the DNDP. We further show how to accelerate the run time of the Slave Problem.

The remainder of this paper is structured as follows. In Section 2, we introduce the general Linear Bilevel Problem and our algorithm. Section 3 introduces the bilevel formulation of the DNDP and shows the linearization. In Section 4, we evaluate the performance of the algorithm on several instances and end with a summary of the proposed procedure, results and outline some future research opportunities.

2. Bilevel Problem and Algorithm

Section 2.1 shows the transformation of the DCBLP into a single-level linear MIP and Benders Decomposition is applied in Section 2.2.

2.1. Transformation to a single-level problem

In the following, we introduce the general formulation for the DCBLP. The leader variables are given by y_i for all $i \in I$ with I the corresponding set of indices and the follower variables by x_j for all $j \in J$ with J the corresponding set. (1) shows the leader objective function with $f'_i \in \mathbb{R}$ for all $i \in I$ and f_j for all $j \in J \in \mathbb{R}$ the objective coefficients. The follower problem is represented by (2)–(4) with the follower objective function in (2) and the follower objective coefficients $c_j \in \mathbb{R}$ for all $j \in J$. K is the set of follower constraints in (3), where each constraint $k \in K$ is defined by its coefficients $a_{kj} \in \mathbb{R}$ for all $i \in I, a'_{kj} \in \mathbb{R}$ for all $j \in J$ and the right hand side b_k. For simplification, we omit constraints in the leader main problem, but the following transformations can all be applied to the more general formulation. Moreover, we assume the partial cooperation assumption (Dempe, 2002; Bialas and Karwan, 1984) – also called an optimistic DCBLP. This allows the leader to select an optimal follower decision among all optimal follower decisions if there exists more than one.

$$\min_y z_L(y,x) = \sum_{i \in I} f'_i y_i + \sum_{j \in J} f_j x_j \tag{1}$$

$$\text{s.t.} \min_x z_F(x) = \sum_{j \in J} c_j x_j \tag{2}$$

$$\sum_{j\in J} a_{kj}x_j + \sum_{i\in I} a'_{ki}y_i \leqslant b_k \qquad \forall k \in K \tag{3}$$

$$x_j \geqslant 0 \qquad \forall j \in J \tag{4}$$

$$y_i \in \{0,1\} \qquad \forall i \in I \tag{5}$$

As a first step, we reformulate this problem to an equivalent non-linear MIP, which is derived by substituting the follower problem by its Karush–Kuhn–Tucker (KKT) conditions, see Cao and Chen (2006) and Bard (1998). In this single-level model, u_k are the dual variables corresponding to the follower constraints (3), and (9) are the dual constraints corresponding to the primal follower variables x_j. (8) compares the objective value of the primal follower problem on the left side with the objective value of the dual follower problem on the right side. Through the duality theorem, this equation guarantees the optimality of the follower problem while optimizing the leader's objective.

$$\min_y z_L(y,x) = \sum_{i\in I} f'_i y_i + \sum_{j\in J} f_j x_j \tag{6}$$

$$\text{s.t.} \sum_{j\in J} a_{kj}x_j + \sum_{i\in I} a'_{ki}y_i \leqslant b_k \qquad \forall k \in K \tag{7}$$

$$\sum_{j\in J} c_j x_j \leqslant \sum_{k\in K} u_k b_k - \sum_{k\in K}\sum_{i\in I} a'_{ki}u_k y_i \tag{8}$$

$$\sum_{k\in K} a_{kj}u_k \leqslant c_j \qquad \forall j \in J \tag{9}$$

$$u_k \leqslant 0 \qquad \forall k \in K \tag{10}$$

$$x_j \geqslant 0 \qquad \forall j \in J \tag{11}$$

$$y_i \in \{0,1\} \qquad \forall i \in I \tag{12}$$

The non-linear term $u_k y_i$ in (8) can be linearized using the approach used in Cao and Chen (2006) and Farvaresh and Sepehri (2011) and (8) is replaced by the following linear constraints with M being a large positive number:

$$\sum_{j\in J} c_j x_j \leqslant \sum_{k\in K} u_k b_k - \sum_{k\in K}\sum_{i\in I} a'_{ki}\mu_{ki} \tag{13}$$

$$\mu_{ki} \leqslant u_k + M(1-y_i) \qquad \forall i \in I, k \in K \tag{14}$$

$$\mu_{ki} \geqslant u_k \qquad \forall i \in I, k \in K \tag{15}$$

$$\mu_{ki} \geqslant -My_i \qquad \forall i \in I, k \in K \tag{16}$$

$$\mu_{ki} \leqslant 0 \qquad \forall i \in I, k \in K \tag{17}$$

Constraints (14)–(17) ensure that the newly introduced decision variables μ_{ki} take the value 0 if $y_i = 0$ and u_k if $y_i = 1$.

2.2. Benders Decomposition

The structure of the linear MIP derived in the last section allows a solution by Benders Decomposition (Benders, 1962). The basic idea of BD is to decompose the problem into a Master Problem and a Slave Problem and to solve these problems repeatedly. The decision variables are divided into complicating variables, which in our case are the binary variables $(y_i)_{i\in I}$, and a set of easier variables, the continuous $(x_j)_{j\in J}, (u_k)_{k\in K}, (\mu_{ki})_{k\in K, i\in I}$. In each iteration, the Master Problem determines one possible leader decision. This solution is used in the Slave Problem to generate optimality-cuts and a feasible solution or feasibility-cuts, which are added to the Master Problem. The main structure of a BD algorithm is shown in Fig. 1. Because of the linearization, we do not have to apply the Generalized Benders Decomposition (Geoffrion, 1972) and consequently avoid the

Step 0: Initialization: upper bound $UBD = \infty$; set y^* to any feasible solution of y

Step 1: Solve the Slave Problem for $y = y^*$. Let z_S be the current optimal value regarding the Slave Problem and set the upper bound $UBD = \min\{UBD, z_S\}$. If the Slave Problem is bounded, then add an optimality cut to the Master Problem, else add a feasibility cut to the Master Problem.

Step 2: Solve the current Master Problem and save the solution y^*. Let z_M be the current optimal value regarding the Master Problem.

Step 3: If $UBD - z_M < tolerance$ then stop else continue with Step 1.

Fig. 1. Benders decomposition: main structure.

convergence problems for bilinear terms. Sahinidis and Grossmann (1991) showed that the Generalized Benders Decomposition might end in a local optimum or even not in an optimum at all for different starting points.

2.2.1. Dual Slave Problem

The Dual Slave Problem is derived by fixing the decision variables y with y^* and dualizing the linearized single-level problem ((1)–(7), (9)–(12), (13)–(17)). In this model, the dual variables $\alpha_k, \beta, \gamma_j, \delta_{1ki}, \delta_{2ki}$ and δ_{3ki} correspond to the constraints (7), (13), (9), (14)–(16) and the dual constraints (19)–(21) to the variables x_j, u_k and μ_{ki}.

$$\max_{\alpha,\beta,\gamma,\delta}\sum_{k\in K}\left(b_k-\sum_{i\in I}a'_{ki}y_i^*\right)\cdot\alpha_k + \sum_{j\in J}c_j\gamma_j+\sum_{i\in I}\sum_{k\in K}(M(1-y_i^*)\delta_{1ki}-My_i^*\delta_{3ki}) \tag{18}$$

$$\text{s.t.}\sum_{k\in K}a_{kj}\alpha_k+c_j\beta\leqslant f_j \qquad \forall j\in J \tag{19}$$

$$-b_k\beta+\sum_{j\in J}a_{kj}\gamma_j-\sum_{i\in I}(\delta_{1ki}+\delta_{2ki})\geqslant 0 \qquad \forall k\in K \tag{20}$$

$$a'_{ki}\beta+\delta_{1ki}+\delta_{2ki}+\delta_{3ki}\geqslant 0 \qquad \forall i\in I,k\in K \tag{21}$$

$$\alpha_k\leqslant 0 \qquad \forall k\in K \tag{22}$$

$$\beta\leqslant 0 \tag{23}$$

$$\gamma_j\leqslant 0 \qquad \forall j\in J \tag{24}$$

$$\delta_{1ki}\leqslant 0 \qquad \forall i\in I,k\in K \tag{25}$$

$$\delta_{2ki},\delta_{3ki}\geqslant 0 \qquad \forall i\in I,k\in K \tag{26}$$

If this problem is feasible with a solution $\alpha^*, \beta^*, \gamma^*$ and δ^*, we add an optimality cut

$$\sum_{i\in I}f'_iy_i+\sum_{k\in K}\left(b_k-\sum_{i\in I}a'_{ki}y_i\right)\cdot\alpha_k^*+\sum_{j\in J}c_j\gamma_j^*+\sum_{i\in I}\sum_{k\in K}(M(1-y_i)\delta_{1ki}^*-My_i\delta_{3ki}^*)\leqslant z \tag{27}$$

to the Master Problem. If it is unbounded, a new constraint, which bounds the objective function with a Big-M M_2, is added to the Slave Problem. M_2 has to be large enough such that no extreme point is cut off and only the extreme rays are bounded. The Slave problem can then be solved by including this constraint:

$$\sum_{k\in K}\left(b_k-\sum_{i\in I}a'_{ki}y_i\right)\cdot\alpha_k+\sum_{j\in J}c_j\gamma_j+\sum_{i\in I}\sum_{k\in K}(M(1-y_i)\delta_{1ki}-My_i\delta_{3ki})\leqslant M_2 \tag{28}$$

The solution of this problem generates the following feasibility cut, which can be added to the Master Problem:

$$\sum_{k\in K}\left(b_k-\sum_{i\in I}a'_{ki}y_i\right)\cdot\alpha_k^*+\sum_{j\in J}c_j\gamma_j^*+\sum_{i\in I}\sum_{k\in K}(M(1-y_i)\delta_{1ki}^*-My_i\delta_{3ki}^*)\leqslant 0 \tag{29}$$

This Slave Problem only contains continuous variables and is easy to solve.

2.2.2. Master problem

Let C_O be the set of solutions $(\alpha^*, \gamma^*, \delta^*)$ of optimality cuts and C_F be the set of solutions $(\alpha^*, \gamma^*, \delta^*)$ of feasibility cuts. In each iteration of the BD, a cut based on the solution of the Slave Problem is added to the respective set. Then, the corresponding Master Problem is defined as follows:

$$\min z \tag{30}$$

$$\text{s.t.}\, z\geqslant\sum_{i\in I}f'_iy_i+\sum_{k\in K}\left(b_k-\sum_{i\in I}a'_{ki}y_i\right)\cdot\alpha_k^*+\sum_{j\in J}c_j\gamma_j^*+\sum_{i\in I}\sum_{k\in K}(M(1-y_i)\delta_{1ki}^*-My_i\delta_{3ki}^*) \quad \forall(\alpha^*,\gamma^*,\delta^*)\in C_O \tag{31}$$

$$0\geqslant\sum_{k\in K}\left(b_k-\sum_{i\in I}a'_{ki}y_i\right)\cdot\alpha_k^*+\sum_{j\in J}c_j\gamma_j^*+\sum_{i\in I}\sum_{k\in K}(M(1-y_i)\delta_{1ki}^*-My_i\delta_{3ki}^*) \quad \forall(\alpha^*,\gamma^*,\delta^*)\in C_F \tag{32}$$

Through the decomposition we have two smaller subproblems which can be solved much faster: the continuous Slave Problem and the usually rather small linear mixed-integer Master Problem.

2.2.3. Acceleration for Slave Problem

Let z_F^* being the optimal objective value of the follower problem (2)–(4) for a fixed y^*. As the dual variables of the Karush–Kuhn–Tucker conditions u_k, μ_{ki} only ensure optimality of the follower problem, they don not appear in the leader objective function (6) and the primal Slave Problem can be expressed as follows:

$$\min_y z_L(y,x) = \sum_{i \in I} f'_i y_i^* + \sum_{j \in J} f_j x_j \tag{33}$$

$$\text{s.t.} \sum_{j \in J} a_{kj} x_j + \sum_{i \in I} a'_{ki} y_i^* \leqslant b_k \qquad \forall k \in K \tag{34}$$

$$\sum_{j \in J} c_j x_j \leqslant z_F^* \tag{35}$$

$$x_j \geqslant 0 \qquad \forall j \in J \tag{36}$$

Having the optimal solution value of the follower objective, (35) ensures that the leader objective is minimized under the condition that the follower objective is minimal. The dual variables α_k and β can be calculated by this formulation, as they are not effected by constraint (9). Afterwards, the dual of the follower problem is solved:

$$\max \sum_{k \in K} u_k b_k - \sum_{k \in K} \sum_{i \in I} a'_{ki} \mu_{ki} \tag{37}$$

$$\text{s.t.} \sum_{k \in K} a_{kj} u_k \leqslant c_j \qquad \forall j \in J \tag{38}$$

$$(14)\text{–}(17) \tag{39}$$

$$u_k \leqslant 0 \qquad \forall k \in K \tag{40}$$

Let $\gamma'_j, \delta'_{1ki}, \delta'_{2ki}$ and δ'_{3ki} be the dual variables of this problem. As the dual of the follower problem does not influence the leader objective function directly but only (37) which is z_F^* in (35), the dual variables of the Slave Problem can be calculated as follows: $\gamma_j = \gamma'_j \beta, \delta_{1ki} = \delta_{1ki}\beta, \delta_{2ki} = \delta_{2ki}\beta$ and $\delta_{3ki} = \delta_{3ki}\beta$. As γ'_j is the shadow price of constraint (38) for (37) which is further the shadow price for (35) and as β is the shadow price of (35) for the leader objective function, $\gamma'_j \beta$ is the shadow price of constraint (38) for the leader objective function. Furthermore, if the calculation in the second step ends in $\beta = 0$, the dual follower problem does not have to be solved.

3. Discrete Network Design Problem

A general definition of the DNDP is given in Section 3.1 and a continuous variable based linearization of a convex function is given in Section 3.2.

3.1. Problem definition

In the DNDP (LeBlanc, 1975; Gao et al., 2005; Poorzahedy and Turnquist, 1982), an existing transportation network is modeled as a set of nodes N, representing origins and destinations or intersections. The nodes are connected via a set of arcs A, which represents the road system. Every arc $a \in A$ is specified by a travel time function $t_a(x) := T_a\left(1 + B_a\left(\frac{x}{c_a}\right)^4\right)$ (Bureau of Public Roads, 1964). T_a is the free flow travel time, B_a the congestion influence parameter and c_a the capacity limit. The demand in the network is represented by the Origin–Destination-Matrix (OD-Matrix). The set of origins is defined as $R \subset N$ and the set of destinations as $S \subset N$. The OD-Matrix is then defined by the values q_{rs}, which is the number of travelers from $r \in R$ to $s \in S$. The set of arcs is divided into two subsets: A_1 is the set of already existing roads, and A_2 the set of possible new roads, which each would cost b_a to build.

The decision maker (leader) has to decide which of the possible new roads of the networks to build subject to a budget B. These decisions will be the binary variables y_a, which are 1 if a new route $a \in A_2$ is built and 0 if not. The leader's objective is to avoid congestion and minimize the total travel time in the network, which is called the system-optimum. The follower, the travelers through the network, minimize their own travel time, which is based on Wardrop's first principle (Wardrop, 1952). This optimum is called the user-optimum. The Paradox of Braess (Braess et al., 2005) showed why these two optima are not necessarily the same and so the formulation as Bilevel Problem is necessary. The flow on each arc $a = (i,j) \in A$ will be the continuous decision variable x_a and the flow for each destination s on arc a the continuous variables $f_a^s = f_{ij}^s$.

The follower problem, which is an uncapacitated Traffic Assignment Problem (TAP) (Nguyen, 1974), can now be formulated for a fixed leader decision y_a^* as follows (Poorzahedy and Turnquist, 1982):

$$\min \sum_{a \in A} \int_0^{x_a} t_a(x)dx = \min \sum_{a \in A} \left(T_a x_a + \frac{T_a B_a}{5 c_a^4} x_a^5 \right) \tag{41}$$

$$\text{s.t.} \sum_{j \in N} f_{kj}^s - \sum_{i \in N} f_{ik}^s = q_{ks} \qquad \forall s \in S, k \in N \tag{42}$$

$$x_a = \sum_{s \in S} f_a^s \qquad \forall a \in A \tag{43}$$

$$x_a \leqslant M_3 y_a^* \qquad \forall a \in A_2 \tag{44}$$

$$f_{ij}^s \geqslant 0 \qquad \forall s \in S, (i,j) \in A \tag{45}$$

Constraint (42) is the flow conservation constraint for each destination node $s \in S$ and each node $k \in N$. This constraint ensures that all flows with destination s, which flow into node k, and the demand of node k with destination s have to flow out of node k. In (43) the aggregated flow on a link is computed and (44) ensures that only roads which are built can be used. The Big-M M_3 has to be larger than the maximum possible flow of the network.

The leader problem, which only consists of the objective function with a budget constraint, is:

$$\min \sum_{a \in A} x_a t_a(x_a) \tag{46}$$

$$\text{s.t.} \sum_{a \in A_2} b_a y_a \leqslant B \tag{47}$$

$$y_a \in \{0,1\} \qquad \forall a \in A_2 \tag{48}$$

The objective functions of both problems each contain a linear term and a non-linear term of the form x^5. For the non-linear terms, we use the following piecewise linear approximation, which only requires continuous auxiliary variables.

3.2. Linearization of non-linear convex functions

Let $f(x)$ be an increasing, convex and non-linear function. Assume $m+1$ approximation points $(\vartheta_0, f_0), (\vartheta_1, f_1), \ldots, (\vartheta_m, f_m)$ with $f_i := f(\vartheta_i)$. Further, $f(x)$ is a function in the single flow variable x_a and $\vartheta_m \geqslant \max_{a \in A} x_a$ has to hold. The trivial upper bound is $\sum_{r \in R, s \in S} q_{rs}$. However, as x_a can be much smaller than $\sum_{r \in R, s \in S} q_{rs}$, empirical upper bounds can further improve the quality of the approximation. Define $a_i := \frac{f_i - f_{i-1}}{\vartheta_i - \vartheta_{i-1}}$ and $b_i := -\vartheta_{i-1} a_i + f_{i-1}$. Then $f(x)$ can be approximated by the following piecewise linear function:

$$\bar{f}(x) := \begin{cases} a_i x + b_i, & \text{for } x \in [\vartheta_{i-1}, \vartheta_i), i = 1, \ldots, m-1 \\ a_i x + b_i, & \text{for } x \in [\vartheta_{i-1}, \infty), i = m \end{cases} \tag{49}$$

It is clear that $a_i - a_{i-1} \geqslant 0$ and Nemhauser and Wolsey (1988) stated that no binary variables are needed for the approximation.

Instead, $\bar{f}(x)$ can be minimized by the following LP:

$$\min\ f_0 + a_1 x_1 + \sum_{i=2}^{m} (a_i - a_{i-1}) x_i \tag{50}$$

$$\text{s.t. } x_1 \leqslant x_i + \vartheta_{i-1} \quad i = 2, \ldots, m \tag{51}$$

$$x_i \geqslant 0 \qquad i = 1, \ldots, m \tag{52}$$

As in each (ϑ_i, f_i) a new slope a_i starts, we have to add $(a_i - a_{i-1})x_i$ from that point on with $x_i = x_1 - \vartheta_{i-1}$, but do not subtract anything if $x_1 < \vartheta_{i-1}$. Because of the minimization problem and the definition of the objective function constraints, (51) and (52) ensure that x_i takes the value of $\min\{0, x_1 - \vartheta_{i-1}\}$ and the defined optimization problem minimizes $f(x)$. In the example of Fig. 2, $x_1 > \vartheta_1$ and we have to add $(a_2 - a_1)x_2$ with $x_2 = (x_1 - \vartheta_1)$ (gray area), but $x_3 = 0$.

Imamoto and Tang (2008) proposed a recursive algorithm to find the optimal minimax solution of the piecewise linear approximation of convex functions. In the preprocessing, this algorithm optimizes the x-coordinates such that the maximal approximation error between two adjacent points is minimized.

Applying this piecewise linear approximation to the non-linear objective functions of the DNDP transforms the problem into a Linear Bilevel Problem without violating the convexity of the objective functions. Therefore, the user equilibrium keeps its unique solution (Sheffi, 1985) and the KKT-conditions in Section 2 are sufficient (Bard, 1998).

4. Numerical tests

To show the efficiency of our approach, we used three different examples: a small example (S) taken from Gao et al. (2005), the well-known Sioux-Fall network (M) of LeBlanc (1975) and – as a large scale example – the network of Berlin Mitte Center (L) (Bar-Gera, 2013). In the latter two networks, we used the data of the TAP and added possible new streets to the system in order to have large but also more complex instances, as the complexity also is caused by the number of possible new roads. LeBlanc (1975), Gao et al. (2005) and Luathep et al. (2011) solved the Sioux-Fall network only for 5 possible new roads and (L) was never considered as a DNDP problem. In the small example, the costs for building all 6 new roads are 70, in (M) we added 10 (15) new roads with total costs of 110 (155) and in the large example, we added 10 potential new links with

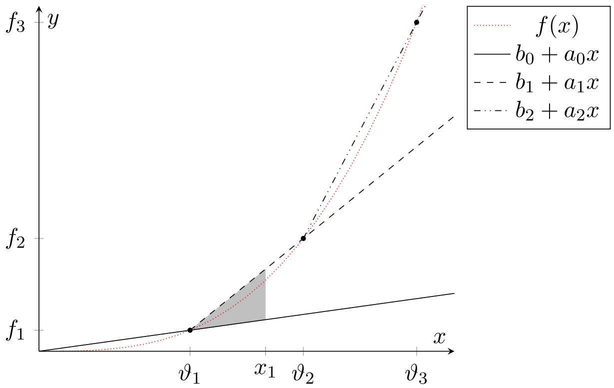

Fig. 2. Example of an approximation of $f(x)$ with 4 data points.

Table 1
Key information of examples.

	#Nodes	#OD nodes	#Arcs	#OD pairs	Total flow	#New arcs	Total costs	ϑ_m
S	12	2	23	1	20	6	70	11
M	24	24	86	528	3606	10 (15)	110 (155)	260
L	398	36	871	1260	11481.92	10	180	1400

total building costs of 180. As empirical upper bounds for ϑ_m were not available, we solved the TAP in a preprocessing with several approximations until $\vartheta_m \geqslant \max_{a \in A} x_a$ was satisfied. The key information on the network sizes is given in Table 1.

The reported objective value was calculated by evaluating the non-linear objective function of the leader with the solution obtained from the linearization approximation.

The tests were performed on an Intel(R) Core(TM) i7–2640 M CPU, 2.8 GHz, 4 GB RAM and implemented in Xpress-MP 7.3. Technically, we solved the primal formulations in the Slave Problem of the BD and got the dual variables through the Xpress functions because the tests showed that the solution time of the primal formulation in our problem is faster than the dual formulation. The calculations were done with several different budgets B and different numbers of approximation points m as introduced in Section 3.2. Tables 2–5 show the computational results. The performance of the algorithm was measured by the run time (*time*) and the number of iterations (*iter*) of the BD algorithm. The calculation time of (M) and (L) were distinguished between the total run time (*time total*) and the solution time for the first Slave Problem (*time step 1*), which shows the complexity of the TAP. Furthermore, the single-level formulation for (M) was solved to compare the

Table 2
Results for example (S).

B	m	Time (sec)	Iter	Obj BD	Obj Gao	GAP (%)
10	10	0.03	3	4088.28	4076.59	0.29
	20	0.03	3	4075.37	4076.59	−0.03
	40	0.06	3	4075.00	4076.59	−0.04
20	10	0.09	8	3959.99	3952.53	0.19
	20	0.12	8	3944.64	3952.53	−0.20
	40	0.17	8	3952.20	3952.53	−0.01
30	10	0.29	12	2754.68	2668.58	3.23
	20	0.30	12	2678.22	2668.58	0.36
	40	0.60	12	2677.94	2668.58	0.35
40	10	0.47	19	2560.18	2524.59	1.41
	20	1.15	19	2520.03	2524.59	−0.18
	40	1.06	19	2527.47	2524.59	0.11
50	10	1.13	23	2397.81	2404.82	−0.29
	20	1.34	23	2406.67	2404.82	0.08
	40	1.42	23	2397.94	2404.82	−0.29
60	10	1.62	28	2314.27	2281.73	1.43
	20	1.85	28	2286.88	2281.73	0.23
	40	2.19	28	2281.58	2281.73	−0.01
70	10	1.68	32	2289.24	2256.96	1.43
	20	1.91	32	2259.99	2256.96	0.13
	40	1.71	32	2255.37	2256.96	−0.07

Table 3
Results for example (M) with 10 new links.

B	m	BD time (sec)		Iter	SL time	Opt found (sec)		GAP_A
		step 1	total		(sec)	BD	SL	(in %)
20	20	0.15	1.63	16	9.15	0.20	3.12	1.739
	40	0.27	2.78	16	21.99	0.35	8.95	1.239
	80	0.64	4.88	15	52.09	0.65	50.99	0.293
	140	1.24	8.89	16	82.35	1.11	78.43	0.098
40	20	0.13	6.27	40	13.46	1.25	6.33	1.461
	40	0.26	11.21	47	19.85	1.91	3.80	0.641
	80	0.63	22.42	51	46.82	4.40	42.43	0.256
	140	1.31	37.93	52	59.38	5.11	56.78	0.055
60	20	0.14	8.41	59	19.22	2.14	19.05	3.086
	40	0.27	14.01	61	16.77	3.45	15.69	1.582
	80	0.69	24.27	60	22.97	6.88	22.49	0.345
	140	1.28	39.88	59	48.70	11.49	47.43	0.073
80	20	0.15	2.24	27	8.05	1.08	1.98	3.097
	40	0.27	4.80	31	14.05	2.94	14.03	1.427
	80	0.64	7.90	26	26.05	5.16	22.91	0.394
	140	1.41	17.80	29	49.10	12.27	48.19	0.083
100	20	0.14	2.51	32	10.46	0.16	10.45	2.974
	40	0.27	4.56	29	6.47	0.31	6.45	1.267
	80	0.65	10.29	34	20.73	0.61	19.98	0.352
	140	1.30	13.88	24	29.72	1.16	28.41	0.125
120	20	0.14	0.26	2	2.05	0.26	2.05	2.745
	40	0.27	0.51	2	8.46	0.51	8.44	0.920
	80	0.65	1.10	2	16.90	1.10	16.85	0.317
	140	1.25	2.86	2	13.07	2.86	13.04	0.127

Table 4
Results for example (M) with 15 new links.

B	m	BD time (sec)		Iter	SL time	Opt found (sec)		GAP_A
		step 1	total		(sec)	BD	SL	(in %)
20	20	0.16	3.40	23	8.69	0.30	4.47	1.739
	40	0.32	8.47	27	68.98	0.63	68.45	1.239
	80	1.29	16.64	26	154.04	1.28	150.26	0.293
	140	1.87	19.05	27	417.73	1.41	409.79	0.098
40	20	0.19	24.14	82	14.16	0.59	10.75	2.599
	40	0.37	31.25	82	58.87	0.76	37.73	1.204
	80	0.68	48.34	86	182.49	1.12	180.63	0.165
	140	1.32	76.26	86	375.84	1.77	337.72	0.105
60	20	0.19	19.92	74	14.15	2.15	13.29	2.302
	40	0.38	42.43	86	43.59	4.44	35.49	1.091
	80	1.08	62.50	81	164.84	5.40	156.90	0.280
	140	1.48	92.04	98	399.12	22.54	397.48	0.136
80	20	0.23	20.65	73	19.44	1.70	19.28	5.286
	40	0.45	20.71	59	45.80	4.21	45.62	1.377
	80	0.66	47.49	82	132.34	11.58	130.98	0.488
	140	1.54	59.83	76	326.61	33.85	303.80	0.161
100	20	0.19	8.17	46	15.56	3.38	15.53	5.122
	40	0.33	24.46	89	59.61	2.20	59.58	1.243
	80	0.80	29.10	67	133.41	13.46	133.13	0.519
	140	1.30	72.32	92	339.39	23.58	338.28	0.109
120	20	0.17	6.81	65	16.59	5.66	16.47	4.979
	40	0.30	13.84	66	50.58	9.64	50.32	1.108
	80	0.95	36.21	93	144.01	26.87	141.07	0.543
	140	1.29	61.25	88	453.26	33.41	453.09	0.097
140	20	0.17	3.62	38	12.07	1.33	12.05	5.081
	40	0.31	11.77	55	48.98	3.21	48.93	1.079
	80	0.80	15.11	36	134.50	5.88	132.79	0.473
	140	1.30	49.70	73	389.57	12.25	386.08	0.077
160	20	0.19	0.35	2	3.87	0.35	3.86	5.348
	40	0.34	0.97	2	33.05	0.97	33.03	1.084
	80	0.79	2.01	2	117.43	2.01	117.40	0.467
	140	1.30	42.13	64	323.60	1.32	323.53	0.113

Table 5
Results for example (L).

B	m	Time		Iter	GAP_A
		Step 1 (sec)	Total (min)		(in %)
30	40	22	6	17	0.251
	80	85	22	17	0.045
	140	154	43	17	0.017
60	40	25	23	63	0.250
	80	84	109	79	0.053
	140	180	218	79	0.018

run time of the single-level formulation (*SL time*) and the run time for finding the optimal solution in the BD (*opt found BD*) and the single-level formulation (*opt found SL*).

For (S), the leader objective value of the BD algorithm (*obj BD*) was compared with the leader objective value of Gao (*obj Gao*) and the GAP between both values (*GAP*) gives a measure for the quality of the approximation. As no optimal solution for (M) and (L) is reported in the literature, we used the GAP between the approximated objective value and the value of the evaluated non-linear objective function (GAP_A) as a quality measure.

The results show that the run time of the algorithm depends on several factors: More approximation points m and the size of the network increase the run time. Increasing the number of possible new links, which we did in Table 4 compared to Table 3, increases the number of iterations accompanied with an increased run time. Furthermore, for (M) and (L) the number of iterations and run time increases by increasing the budget B up to 50% of the total costs but starts to decrease again. In the small example, the GAP to the optimal solution is small in all cases and the algorithm ends in the optimal leader decision in all instances. In (M) and (L), the approximation GAP GAP_A decreases by increasing the approximation points but the optimal leader decision does not change for the different approximations.

Furthermore, one can see in Table 2 for the small example that the leader objective is oscillating around the objective value by Gao et al. (2005), because we are not only approximating the leader objective, but also the follower objective, which means we are approximating the feasible region of the leader problem. However, further tests showed that this effect is highly dependent on the test instance. Adding additional flows or arcs to the network already reduced this effect for some budgets.

The slower run time for larger networks and more approximation points is related to the larger Traffic Assignment Problem to be solved. As in (L) even the TAP is difficult to solve, the run time is much higher. The number of iterations comes from the complexity of the problem: More possible links enlarge the solution space, but if the budget is small, one only builds 1 or 2 roads and the number of possibilities is small. If the budget is large, almost every road can be built and the decision is easier.

For further acceleration, we applied pareto-optimal cuts as proposed in Magnanti and Wong (1981) to reduce the number of iterations. However, the effort of calculating these cuts could not reduce the total run time. The reason is the increased effort of solving the Slave Problem.

Besides the comparison with the single level formulation we also compared our algorithm with the algorithm of Bard and Moore (1990). Our algorithm finds the optimal solution significantly faster. Even the case with $m = 10$ and $B = 10$ in (S) could only be solved in 36 s and 881 iterations by the algorithm of Bard and Moore (1990). This is related to the large number of follower variables relative to the number of leader variables. In fact, in the small example, only 2 instances could be solved in less than 1 min and 50% could not be solved in 1 h. Tables 3 and 4 show that the Benders Decomposition reduced the run time on average by 61% and 63%. Moreover, the run time for finding the optimal solution is reduced on average by 84% and 92%. Even though the BD was slightly slower in a few cases, the optimal solution is still found significantly faster.

Finally, we can state that the algorithm always ended in the same decision for building roads without depending on the accuracy of the approximation. This means that the approximation GAP_A of 2% did not influence the leader decision: Moreover, the results of all instances in Tables 3–5 show that as few as 10, 20 resp. 40 approximation points gave a good approximation and the effort of the better approximation does not improve results; but increases run times.

5. Conclusions

We proposed a Benders Decomposition method for solving DCBLPs to global optimality. Benders Decomposition is applied to a reformulation of the DCBLP to a single-level MIP and the Slave Problem solving is further accelerated. The computational results show that this approach, to the best of our knowledge, is the first algorithm solving such large instances for DCBLP and DNDP, even though we just used a basic BD. As the solution time of the Slave Problem, compared to the Master Problem, is rather large, the acceleration of solving this problem is of interest for future research.

Our approach enhances computational capabilities of existing (discrete) network design approaches. However, at the same time it has several limitations which are venues for future research. These are given demand matrices rather than elastic demands and road pricing models, discrete potential new roads with given capacity rather than continuous network design problems for large urban networks, and deterministic rather than stochastic user equilibria.

Acknowledgments

We would like to thank the editor and two anonymous referees for their helpful comments to improve the manuscript.

References

Abdulaal, M., LeBlanc, L.J., 1979. Continuous equilibrium network design models. Transportation Research Part B 13 (1), 19–32.
Bard, J.F., 1998. Practical Bilevel Optimization: Algorithms and Applications. Springer.
Bard, J., Moore, J., 1990. A branch and bound algorithm for the bilevel programming problem. SIAM Journal on Scientific and Statistical Computing 11 (2), 281–292.
Bar-Gera, H. (2013). Transportation Network Test Problems. url: <http://www.bgu.ac.il/∼bargera/tntp/>.
Ben-Ayed, O., Blair, C.E., 1990. Computational difficulties of bilevel linear programming. Operations Research 38 (3), 556–560.
Benders, J., 1962. Partitioning procedures for solving mixed-variables programming problems. Numerische Mathematik 4 (1), 238–252.
Bialas, W.F., Karwan, M.H., 1984. Two-level linear programming. Management Science 30 (8), 1004–1020.
Braess, D., Nagurney, A., Wakolbinger, T., 2005. On a paradox of traffic planning. Transportation Science 39 (4), 446–450.
Bureau of Public Roads, 1964. Traffic Assignment Manual. U.S. Department of Commerce, Urban Planning Division. Washington, DC.
Cao, D., Chen, M., 2006. Capacitated plant selection in a decentralized manufacturing environment: a bilevel optimization approach. European Journal of Operational Research 169 (1), 97–110.
Colson, B., Marcotte, P., Savard, G., 2005. Bilevel programming: a survey. 4OR 3 (2), 87–107.
Dempe, S., 2002. Foundations of Bilevel Programming. Springer.
Ekström, J., Sumalee, A., Lo, H.K., 2012. Optimizing toll locations and levels using a mixed integer linear approximation approach. Transportation Research Part B 46 (7), 834–854.
Farahani, R.Z., Miandoabchi, E., Szeto, W., Rashidi, H., 2013. A review of urban transportation network design problems. European Journal of Operational Research 229 (2), 281–302.
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 (5), 623–640.
Gao, Z., Wu, J., Sun, H., 2005. Solution algorithm for the bi-level discrete network design problem. Transportation Research Part B 39 (6), 479–495.
Geoffrion, A.M., 1972. Generalized benders decomposition. Journal of Optimization Theory and Applications 10 (4), 237–260.
Grossmann, I.E., Floudas, C.A., 1987. Active constraint strategy for flexibility analysis in chemical processes. Computers and Chemical Engineering 11 (6), 675–693.
Hansen, P., Jaumard, B., Savard, G., 1992. New branch-and-bound rules for linear bilevel programming. SIAM Journal on Scientific and Statistical Computing 13 (5), 1194–1217.
Imamoto, A., Tang, B., 2008. Optimal piecewise linear approximation of convex functions. In: Proceedings of the World Congress on Engineering and Computer Science. Citeseer, pp. 1191–1194.
LeBlanc, L.J., 1975. An algorithm for the discrete network design problem. Transportation Science 9 (3), 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 45 (5), 808–827.
Magnanti, T.L., Wong, R.T., 1981. Accelerating benders decomposition: algorithmic enhancement and model selection criteria. Operations Research 29 (3), 464–484.
Mathew, T.V., Sharma, S., 2009. Capacity expansion problem for large urban transportation networks. Journal of Transportation Engineering 135 (7), 406–415.
Meng, Q., Yang, H., 2002. Benefit distribution and equity in road network design. Transportation Research Part B 36 (1), 19–35.
Nemhauser, G.L., Wolsey, L.A., 1988. Integer and Combinatorial Optimization. Wiley, New York.
Nguyen, S., 1974. An algorithm for the traffic assignment problem. Transportation Science 8 (3), 203–216.
Poorzahedy, H., Abulghasemi, F., 2005. Application of ant system to network design problem. Transportation 32 (3), 251–273.
Poorzahedy, H., Rouhani, O.M., 2007. Hybrid meta-heuristic algorithms for solving network design problem. European Journal of Operational Research 182 (2), 578–596.
Poorzahedy, H., Turnquist, M.A., 1982. Approximate algorithms for the discrete network design problem. Transportation Research Part B 16 (1), 45–55.
Saharidis, G., Ierapetritou, M., 2009. Resolution method for mixed integer bi-level linear problems based on decomposition technique. Journal of Global Optimization 44 (1), 29–51.
Sahinidis, N., Grossmann, I.E., 1991. Convergence properties of generalized benders decomposition. Computers and Chemical Engineering 15 (7), 481–491.
Schneeweiß, C., 2003. Distributed Decision Making, 2nd ed. Springer.
Sheffi, Y., 1985. Urban Transportation Networks: Equilibrium Analysis with Mathematical Programming Methods. Prentice-Hall, Inc.
Suh, S., Kim, T., 1992. Solving nonlinear bilevel programming models of the equilibrium network design problem: a comparative review. Annals of Operations Research 34 (1), 203–218.
Wang, S., Meng, Q., Yang, H., 2013. Global optimization methods for the discrete network design problem. Transportation Research Part B 50, 42–60.
Wardrop, J.G., 1952. Some theoretical aspects of road traffic research. ICE Proceedings: Engineering Divisions, vol. 1.3. Thomas Telford, pp. 325–362.
Xu, T., Wei, H., Hu, G., 2009. Study on continuous network design problem using simulated annealing and genetic algorithm. Expert Systems with Applications 36 (2), 1322–1328.