A cutting-plane algorithm for the continuous network design problem based on value function cuts and outer approximation

Michael W. Levin*[1] and David Rey[2]

[1]Department of Civil, Environmental, and Geo- Engineering, University of Minnesota, United States
[2]SKEMA Business School, Université Côte d'Azur, France

Short summary

Transportation network design, or the problem of optimizing infrastructure for a societal goal, subject to individual travelers optimizing their behavior for their own preferences arises frequently in many contexts. However, it is also an NP-hard problem due to the leader-follower or bi-level structure involving a follower objective that is different from yet significantly affects the leader objective. Creating exact algorithms has been particularly difficult for the continuous network design problem (CNDP), in which leader variables are continuous, because of the challenges in solving a mathematical program with explicit constraints for follower optimality. We present an exact algorithm for CNDP based on using the high-point relaxation (system optimal CNDP, or CNDP without the user equilibrium constraint) to find lower bounds and solving the traffic assignment follower problem to find upper bounds on the optimal solution. To solve the high-point relaxation faster, we outer-approximate the objective and value function cuts and use column generation to obtain a sequence of linear programs that can be solved relatively quickly. Compared to prior work on exact methods for CNDP, we can find exact solutions for the same small test networks in much less time, or solve CNDP on much larger networks than have been solved in the literature.

Keywords: continuous network design problem; outer approximation; value function cuts; traffic assignment; user equilibrium

1 Introduction

The transportation network design problem (NDP) (Farahani et al., 2013) is the problem of deciding where to improve road network infrastructure so as to optimize some societal goal such as minimizing traffic congestion. The difficulty in solving NDP comes from the fact that the effects of new infrastructure depend on drivers' behavior, and drivers usually behave according to their own preferences such as minimizing their individual travel times. We focus on the continuous NDP (CNDP), in which design variables can take on continuous values.

Problem definition

The standard CNDP is defined as follows. Consider a traffic network $\mathcal{G} = (\mathcal{N}, \mathcal{A})$ with nodes $\mathcal{N}$ and links $\mathcal{A}$. The trips from r to s is d_{rs}. Some (possibly all) links can be modified by adding extra capacity. The problem is to decide y_a, the extra capacity added to link a, so as to minimize total congestion. We assume that link variables y_a for $a \in \mathcal{A}$ are lower-bounded by 0 and upper-bounded by $\overline{y}_a$. If $\overline{y}_a = 0$, then we say that the capacity of link a is fixed. Each link a has a travel time function $t_a(x_a, y_a)$ that indicates the travel time on a as a function of link flow x_a and added capacity y_a. For example, the Bureau of Public Roads (BPR) function is

$$t_a(x_a, y_a) = t_a^{\text{ff}} \left(1 + \alpha_a \left(\frac{x_a}{C_a + y_a}\right)^{\beta_a}\right) \tag{1}$$

We make the following assumptions:

1. Drivers behave according to Wardop's user equilibrium principle (Wardrop, 1952)
2. The cost of adding capacity is assumed to be a convex function $g_a(y_a)$.
3. $x_a t_a(x_a, y_a)$, $\int_0^{x_a} t_a(\omega, y_a) d\omega$, and $\int_0^{x_a} t_a(\omega, y_a) d\omega - \int_0^{x_a^{\text{f}}} t_a(\omega, y_a) d\omega$ are convex. This holds for the BPR travel time function.

For a fixed $\mathbf{y}$, the follower problem is user equilibrium and can be written as

$$\begin{aligned} \texttt{TAP}(\mathbf{y}) = \arg\min_{\mathbf{x}} \quad & B(\mathbf{x}, \mathbf{y}) = \sum_{a \in \mathcal{A}} \int_0^{x_a} t_a(\omega, y_a) d\omega && && \text{(2a)} \\ \text{s.t.} \quad & x_a = \sum_{\pi \in \Pi} \delta_a^\pi h^\pi && \forall a \in \mathcal{A} && \text{(2b)} \\ & \sum_{\pi \in \Pi_{rs}} h^\pi = d_{rs} && \forall (r,s) \in \mathcal{N}^2 && \text{(2c)} \\ & h^\pi \geq 0 && \forall \pi \in \Pi && \text{(2d)} \end{aligned}$$

where π is a path, Π_{rs} is the set of paths from r to s, and $\Pi = \bigcup_{(r,s) \in \mathcal{N}^2} \Pi_{rs}$.

The continuous network design problem (CNDP) is

$$\begin{aligned} \texttt{CNDP} : \min \quad & Z(\mathbf{x}, \mathbf{y}) = \sum_{a \in \mathcal{A}} \left(x_a t_a(x_a, y_a) + g_a(y_a)\right) && && \text{(3a)} \\ \text{s.t.} \quad & 0 \leq y_a \leq \overline{y}_a && \forall a \in \mathcal{A} && \text{(3b)} \\ & \mathbf{x} \in \texttt{TAP}(\mathbf{y}) && && \text{(3c)} \end{aligned}$$

which is a bi-level optimization problem due to Constraint (3c) which requires that user equilibrium conditions hold and represents a $\mathbf{y}$-parameterized traffic assignment problem. Due to the bi-level structure, CNDP is non-convex even if the leader objective and follower objective are convex, and NP-hard to solve (Gairing et al., 2017). Therefore, finding exact solutions is computationally difficult, and researchers who encounter CNDP often use metaheuristics such as genetic algorithms (Nayeem et al., 2014; Arbex & da Cunha, 2015; Levin et al., 2020). Finding exact solutions is beneficial for system performance. Yet, existing exact approaches for the CNDP do not scale well and struggle to compute globally optimal solutions even for problem instances based on small test networks such as Sioux Falls (Du & Wang, 2016; Li et al., 2012).

Our contributions

Consider the limitations of two recent approaches to CNDP.

- Du & Wang (2016) used complementary slackness to ensure bi-level feasibility, and used geometric programming to handle its nonlinearity. However, that approach introduces difficulties with the linear constraints, and requires enumerating all paths.
- Li et al. (2012) does not require path enumeration, but instead solve a sequence of concave minimization problems. Each subproblem is therefore NP-hard. The reported computation times from these recent methods on the 6 node, 16 link network of Harker & Friesz (1984) range from 1.5 minutes using D. Z. Wang & Lo (2010)'s piecewise approximation to 12hrs from Li et al. (2012).

These prior methods involve complex subproblems that are difficult to solve, and do not take advantage of the relatively simple subproblems that CNDP is related to. First, for any fixed leader variable, the follower problem of traffic assignment can be solved efficiently on large networks using dedicated algorithms (e.g. Dial, 2006; Bar-Gera, 2010). Second, in many use cases (e.g. the Bureau of Public Roads travel time function), the high-point relaxation of CNDP has a convex objective and linear constraints. Moreover, value function cuts using the Beckmann et al. (1956) objective for traffic assignment are convex and sufficient to enforce follower optimality. Using outer approximation (Duran & Grossmann, 1986), these convex nonlinear functions can be replaced with a set of linear constraints, yielding a linear program that can be solved quickly even on large networks using column generation. Although prior work made use of outer approximation, their reformulations involved a concave minimization problem (Li et al., 2012) or a MILP for approximating a nonlinear function (Liu & Wang, 2015), limiting their efficiency.

The contributions of this study are as follows: we present a new algorithm for CNDP based on adding value function cuts to the high-point relaxation, and using outer approximation to form a linear approximation of this problem. We achieve global optimality by sequentially obtaining tighter valid upper and lower bounds. As this algorithm involves solving linear programs and traffic assignment, it is far more efficient than existing methods, which we demonstrate on both the Harker & Friesz (1984) network used in prior CNDP work and on much larger networks.

2 Methodology

Our algorithm builds on elements of the literature from both DNDP and CNDP literature. We focus on computing a sequence of progressively tighter upper and lower bounds, like several exact algorithms for DNDP (Leblanc, 1975; Farvaresh & Sepehri, 2013). Upper bounds are computed from traffic assignment, whereas lower bounds are found by solving the high-point relaxation while adding value function constraints like in S. Wang et al. (2013) to move towards follower optimality.

The structure of our proposed algorithm is as follows: SO-CNDP computes an objective $Z^\star_{\text{lb}}$ that is a global lower-bound to the optimal objective value $Z^\star$ because the feasible space has been relaxed. Because SO-CNDP has a convex objective and linear constraints, it is far easier to solve. Meanwhile, for any fixed $\mathbf{y}$, we can obtain an upper bound on the optimal objective by solving $\mathbf{x}^{\text{f}} = \texttt{TAP}(\mathbf{y})$ and computing $Z(\mathbf{x}^{\text{f}}, \mathbf{y})$. It is an upper bound because the fixed $\mathbf{y}$ may not be optimal, but if $\mathbf{y}$ satisfies constraint (3b) then the point $(\mathbf{x}^{\text{f}}, \mathbf{y})$ is a feasible solution to CNDP. If we find a solution $(\mathbf{x}^{\text{l}}, \mathbf{y}^{\text{l}})$ such that

$$Z(\texttt{TAP}(\mathbf{y}^{\text{l}}), \mathbf{y}^{\text{l}}) - Z^\star_{\text{lb}} \leq \epsilon \tag{4}$$

where $Z^\star_{\text{lb}}$ is an optimal solution to the high-point relaxation SO-CNDP, then the solution $\mathbf{y}^{\text{l}}$ must be within ϵ of the globally optimal solution and we can terminate. Of course, $Z^\star_{\text{lb}}$ from SO-CNDP alone will have a substantial gap from $Z(\texttt{TAP}(\mathbf{y}^{\text{l}}), \mathbf{y}^{\text{l}})$ due to the lack of the user equilibrium constraint. Therefore, we need to introduce additional value function cuts on SO-CNDP to improve its accuracy while maintaining its solution as a lower bound to the global optimum of CNDP.

Value function cuts

We start by considering the high-point relaxation of CNDP which is obtained by relaxing Constraint (3c). This relaxation represents a system optimal CNDP (SO-CNDP):

$$\begin{aligned}
\texttt{SO-CNDP} : Z_{\text{lb}} = \min \quad & Z(\mathbf{x},\mathbf{y}) = \sum_{a\in\mathcal{A}} x_a t_a(x_a,y_a) + g_a(y_a) && && \text{(5a)}\\
\text{s.t.} \quad & 0 \le y_a \le \overline{y}_a && \forall a \in \mathcal{A} && \text{(5b)}\\
& x_a = \sum_{\pi\in\Pi} \delta_a^\pi h^\pi && \forall a \in \mathcal{A} && \text{(5c)}\\
& \sum_{\pi\in\Pi_{rs}} h^\pi = d_{rs} && \forall (r,s) \in \mathcal{Z}^2 && \text{(5d)}\\
& h^\pi \ge 0 && \forall \pi \in \Pi && \text{(5e)}
\end{aligned}$$

We reformulate problem (5) with value function cuts to obtain the original CNDP:

Proposition 1. *Consider the high-point relaxation of SO-CNDP augmented with value function cuts:*

$$\begin{aligned}
\min \quad & Z(\mathbf{x},\mathbf{y}) = \sum_{a\in\mathcal{A}} \left(x_a t_a(x_a,y_a) + g_a(y_a)\right) && && \text{(6a)}\\
\text{s.t.} \quad & 0 \le y_a \le \overline{y}_a && \forall a \in \mathcal{A} && \text{(6b)}\\
& x_a = \sum_{\pi\in\Pi} \delta_a^\pi h^\pi && \forall a \in \mathcal{A} && \text{(6c)}\\
& \sum_{\pi\in\Pi_{rs}} h^\pi = d_{rs} && \forall (r,s) \in \mathcal{Z}^2 && \text{(6d)}\\
& h^\pi \ge 0 && \forall \pi \in \Pi && \text{(6e)}\\
& B(\mathbf{x},\mathbf{y}) \le B(\mathbf{x}',\mathbf{y}) && \forall \mathbf{x}' \in \mathcal{X} && \text{(6f)}
\end{aligned}$$

where constraints (6f) *are the value function cuts. Then problem* (6) *is equivalent to problem* (3).

Proof. We show equality of the feasible regions, which is sufficient because the objective functions are identical. Let $(\mathbf{x},\mathbf{y})$ be a point that is feasible for problem (6) with $\mathbf{x} \in \mathcal{X}$. Because it satisfies constraint (6f) it minimizes $B(\mathbf{x},\mathbf{y})$ over link flows $\mathbf{x}$, which makes it bi-level feasible for CNDP. Conversely, consider a point $(\mathbf{x},\mathbf{y})$ that is feasible for problem (3). Then $\mathbf{x} \in \mathcal{X}$ and $B(\mathbf{x},\mathbf{y})$ is minimized by definition of `TAP`, so it satisfies constraint (6f) and is feasible for problem (6). □

The purpose of the value function cuts is to cut out bi-level infeasible solution from SO-CNDP. Since there is an infinite number of value function cuts (6f), we develop a cutting plane algorithm that iteratively adds such cuts to SO-DNDP. To achieve convergence, we need to show that adding cut (7) excludes ranges of $\mathbf{x}$ and $\mathbf{y}$ values, so that we are sequentially cutting a non-finite subset of the feasible region of SO-CNDP. We first verify correctness: after adding cut (7), SO-CNDP still achieves a lower bound on CNDP.

Proposition 2. *After adding cut* (7) *to problem* (5) *for a specific* $\mathbf{x}^{\text{f}} \in \mathcal{X}$*, the optimal solution to problem* (5)*,* $Z^\star_{\text{lb}}$*, is a lower bound on the optimal function value (OFV) of CNDP.*

Proof. We obtain a lower bound because we are adding a subset of the value function cuts in problem (6) that creates an equivalent problem to CNDP. □

Now, we want to show that cut (7) is effective at removing a non-finite subset of the feasible region of $\mathbf{x}$ that would be bi-level infeasible.

Proposition 3. *Suppose SO-CNDP gives a solution* $(\mathbf{x}^1,\mathbf{y}^1)$ *where* $\mathbf{x}^1 \ne \arg\min_{\mathbf{x}} B(\mathbf{x},\mathbf{y}^1)$ *(i.e.* $\mathbf{x}^1$ *is not bi-level feasible). Let* $\mathbf{x}^{\text{f}} = \texttt{TAP}(\mathbf{y}^1)$*. Then adding cut*

$$B(\mathbf{x},\mathbf{y}) \le B\left(\mathbf{x}^{\text{f}},\mathbf{y}\right) \tag{7}$$

excludes $(\mathbf{x}^{\mathrm{l}}, \mathbf{y}^{\mathrm{l}})$ *from the feasible region of problem* (5)*. Furthermore, cut* (7) *excludes an additional range of* $\mathbf{x}$ *values from feasibility in problem* (5)*.*

Proof. Let $\mathbf{x}^{\mathrm{f}} = \arg\min_{\mathbf{x}} B(\mathbf{x}, \mathbf{y}^{\mathrm{l}})$ (which is unique due to convexity of TAP) and $\mathbf{x}^{\mathrm{l}} \neq \mathbf{x}^{\mathrm{f}}$. Then $B(\mathbf{x}^{\mathrm{l}}, \mathbf{y}^{\mathrm{l}}) > B(\mathbf{x}^{\mathrm{f}}, \mathbf{y}^{\mathrm{l}})$ and adding the cut $B(\mathbf{x}, \mathbf{y}) \leq B(\mathbf{x}^{\mathrm{f}}, \mathbf{y})$ to SO-CNDP removes $(\mathbf{x}^{\mathrm{l}}, \mathbf{y}^{\mathrm{l}})$ from the feasible region of problem (5). It also removes any other $\mathbf{x} \neq \mathtt{TAP}(\mathbf{y})$ from feasibility of problem (5). $\square$

Because $\mathbf{x}^{\mathrm{f}}$ is feasible for any given $\mathbf{y}$, once we have some feasible $\mathbf{x}^{\mathrm{f}} \in \mathcal{X}$, constraint (7) is potentially useful for all $\mathbf{y}$. Hence the right hand side of $B(\mathbf{x}^{\mathrm{f}}, \mathbf{y})$ holds for all $\mathbf{y}$. Furthermore, if we consider some $\mathbf{y}' \neq \mathbf{y}$ and the best corresponding $\mathbf{x}'$ from problem (5) defined as

$$\mathbf{x}' \in \underset{\mathbf{x} \in \mathcal{X}: B(\mathbf{x}, \mathbf{y}') \leq B(\mathbf{x}^{\mathrm{f}}, \mathbf{y})}{\arg\min} \quad Z(\mathbf{x}, \mathbf{y}') \tag{8}$$

cut (7) can also exclude some $\mathbf{y}'$ where the only feasible $\mathbf{x}$ (after the value function cuts) have a suboptimal value of $Z(\mathbf{x}, \mathbf{y}')$.

If we sequentially add cuts (7) based on different $(\mathbf{x}^{\mathrm{f}}, \mathbf{y}^{\mathrm{l}})$ to problem (5), we will eventually cut away the values of $\mathcal{X}$ that are feasible for problem (5) but infeasible for problem (3). Furthermore, we always obtain a lower bound on CNDP by Proposition 2, so the OFV of problem (5) gradually augmented with value function cuts forms an increasing sequence of lower bounds.

Let n^{vf} be the number of $\mathbf{x}^{\mathrm{f}}$ points used in value function cuts, labeled $\mathbf{x}^{\mathrm{f}}(i)$ for $i \in 1 \dots n^{\mathrm{vf}}$. The set of $\mathbf{x}^{\mathrm{f}}(i)$ points will be built sequentially over iterations. After combining problem (5) with value function cuts, we obtain the partially augmented problem

$$\begin{aligned}
Z_{\mathrm{lb}} = \min \quad & Z(\mathbf{x}, \mathbf{y}) = \sum_{a \in \mathcal{A}} x_a t_a(x_a, y_a) + g_a(y_a) && && \text{(9a)}\\
\text{s.t.} \quad & 0 \leq y_a \leq \overline{y}_a && \forall a \in \mathcal{A} && \text{(9b)}\\
& x_a = \sum_{\pi \in \Pi} \delta_a^\pi h^\pi && \forall a \in \mathcal{A} && \text{(9c)}\\
& \sum_{\pi \in \Pi_{rs}} h^\pi = d_{rs} && \forall (r,s) \in \mathcal{Z}^2 && \text{(9d)}\\
& h^\pi \geq 0 && \forall \pi \in \Pi && \text{(9e)}\\
& B(\mathbf{x}, \mathbf{y}) \leq B\left(\mathbf{x}^{\mathrm{f}}(i), \mathbf{y}\right) && \forall i \in 1 \dots n^{\mathrm{vf}} && \text{(9f)}
\end{aligned}$$

which we will solve iteratively until convergence with the best upper bound.

However, constraint (7) is typically nonlinear in $\mathbf{y}$. For example, for the BPR function (1), $\mathbf{y}$ non-linearly changes the denominator. Therefore it is more computationally expensive to solve problem (5) with it. Instead, we use outer approximation to convert problem (5) into a sequence of linear programs.

Outer approximation of total system travel time

Observe that both $Z(\mathbf{x}, \mathbf{y})$ and its outer approximation are separable by link. Therefore, instead of using the entire $(\hat{\mathbf{x}}, \hat{\mathbf{y}})$ for the outer approximation, we can instead develop a link-based outer approximation for the individual objective function components for each link a. Let ζ_a be the variable used to represent the contribution of link a in the objective function of CNDP via outer approximation. Replacing $Z(\mathbf{x}, \mathbf{y})$ with approximation variable ζ_a, we obtain

$$Z(\hat{\mathbf{x}}, \hat{\mathbf{y}}) + \nabla_{\mathbf{x}} Z(\hat{\mathbf{x}}, \hat{\mathbf{y}}) \cdot (\mathbf{x} - \hat{\mathbf{x}}) + \nabla_{\mathbf{y}} Z(\hat{\mathbf{x}}, \hat{\mathbf{y}}) \cdot (\mathbf{y} - \hat{\mathbf{y}}) \leq \sum_{a \in \mathcal{A}} \zeta_a \tag{10}$$

as the outer approximation cut with the objective of minimizing $\sum_{a\in\mathcal{A}}\zeta_a$. This cut can be decomposed into link-based cuts of the form:

$$\zeta_a \geq \hat{x}_a t_a(\hat{x}_a,\hat{y}_a) + g_a(\hat{y}_a) + (x_a-\hat{x}_a)\left[\hat{x}_a\frac{dt(\hat{x}_a,\hat{y}_a)}{dx_a} + t_a(\hat{x}_a,\hat{y}_a)\right] + (y_a-\hat{y}_a)\left[\hat{x}_a\frac{dt(\hat{x}_a,\hat{y}_a)}{dy_a} + \frac{dg(\hat{y}_a)}{dy_a}\right] \tag{11}$$

In effect, for every $(\hat{\mathbf{x}},\hat{\mathbf{y}})$ point used to generate an outer approximation cut, we instead get the equivalent of $|\mathcal{A}|$ outer approximation cuts for $Z(\mathbf{x},\mathbf{y})$, multiplicatively. If we have n points for generating outer approximation cuts, $(\hat{\mathbf{x}}(i),\hat{\mathbf{y}}(i))$ for $i=1\ldots n$, then we actually end up with $|\mathcal{A}|^n$ outer approximation cuts because every combination of points, e.g. $(\hat{x}_1(1),\hat{x}_2(2),\ldots)$ is giving a cut on ζ_a. Therefore, the outer approximation process will converge much faster using link-based outer approximation cuts than a single cut on the total objective $Z(\mathbf{x},\mathbf{y})$.

Outer approximation of value function cuts

We rewrite constraint (7) as

$$\Delta B(\mathbf{x},\mathbf{y},\mathbf{x}^{\mathrm{f}}) = B(\mathbf{x},\mathbf{y}) - B\left(\mathbf{x}^{\mathrm{f}},\mathbf{y}\right) \leq 0 \tag{12}$$

which we assume to be convex (it is convex for the BPR function). Its outer approximation is

$$B(\mathbf{x}^{\mathrm{l}},\mathbf{y}^{\mathrm{l}}) - B(\mathbf{x}^{\mathrm{f}},\mathbf{y}^{\mathrm{l}}) + \left[\nabla_{\mathbf{x}}B(\mathbf{x}^{\mathrm{l}},\mathbf{y}^{\mathrm{l}})\right]\cdot\left(\mathbf{x}-\mathbf{x}^{\mathrm{l}}\right) + \left[\nabla_{\mathbf{y}}B(\mathbf{x}^{\mathrm{l}},\mathbf{y}^{\mathrm{l}}) - B\left(\mathbf{x}^{\mathrm{f}},\mathbf{y}^{\mathrm{l}}\right)\right]\cdot\left(\mathbf{y}-\mathbf{y}^{\mathrm{l}}\right) \leq 0 \tag{13}$$

To obtain a tighter approximation, observe that $B(\mathbf{x},\mathbf{y})$ is separable by link. Let η_a be the link-specific outer approximation of $B(\mathbf{x},\mathbf{y})$ for link a, i.e.

$$\eta_a \geq \int_0^{x_a^{\mathrm{l}}} t_a\left(\omega,\mathbf{y}_a^{\mathrm{l}}\right)d\omega + \left(x_a-x_a^{\mathrm{l}}\right)t_a\left(x_a^{\mathrm{l}},y_a^{\mathrm{l}}\right) + \left(y_a-y_a^{\mathrm{l}}\right)\int_0^{x_a^{\mathrm{l}}}\frac{dt_a\left(\omega,y_a^{\mathrm{l}}\right)}{dy_a}d\omega \tag{14}$$

for all $(\mathbf{x}^{\mathrm{l}},\mathbf{y}^{\mathrm{l}})$ points. Then we can obtain a partially link-separated outer approximation of cut (12) using η_a:

$$\sum_{a\in\mathcal{A}}\eta_a - B\left(\mathbf{x}^{\mathrm{f}},\mathbf{y}^{\mathrm{l}}\right) - \sum_{a\in\mathcal{A}}\left(y_a-y_a^{\mathrm{l}}\right)\int_0^{x_a^{\mathrm{f}}}\frac{\partial t_a\left(\omega,y_a^{\mathrm{l}}\right)}{\partial y_a}d\omega \leq 0 \tag{15}$$

To summarize, we have approximated cut (12) with the combination of cuts (14) and (15). This separation makes the approximation stronger. Every iteration, we obtain a new $(\mathbf{x}^{\mathrm{l}},\mathbf{y}^{\mathrm{l}})$ point to make a tighter approximation of $B(\mathbf{x},\mathbf{y})$ as $\sum_{a\in\mathcal{A}}\eta_a$ in $\Delta B\leq 0$.

Cutting-plane algorithm for the CNDP

Let n^{vf} be the number of $(\mathbf{x}^{\mathrm{f}},\mathbf{y}^{\mathrm{l}})$ points used to create the outer approximation for $\Delta B(\mathbf{x},\mathbf{y},\mathbf{x}^{\mathrm{f}})$, and let n^{B} be the number of $(\mathbf{x}^{\mathrm{l}},\mathbf{y}^{\mathrm{l}})$ points used to create the outer approximation of the Beckmann function $B(\mathbf{x},\mathbf{y})$. It is possible to have $n^{\mathrm{B}} > n^{\mathrm{vf}}$, meaning there are more outer approximation cuts to tighten η_a than there are value function cuts using η_a. Also let n^{Z} be the number of $(\mathbf{x}^{\mathrm{l}},\mathbf{y}^{\mathrm{l}})$ points used to create the outer approximation of the leader objective. The final linear program to be iteratively solved is as follows:

Algorithm 1: Cutting-plane algorithm for CNDP

```
1  Set $LB \leftarrow 0$, $UB \leftarrow \infty$, $i \leftarrow 1$, $n^{\mathrm{Z}} \leftarrow 0$, $n^{\mathrm{vf}} \leftarrow 0$, $n^{\mathrm{B}} \leftarrow 0$.
2  while $UB - LB > \epsilon$ do
3      Solve SO-CNDP problem (16) to obtain $Z_{\mathrm{lb}}(i)$, $\mathbf{x}^{\mathrm{l}}(i)$, $\mathbf{y}^{\mathrm{l}}(i)$. Set $LB \leftarrow Z_{\mathrm{lb}}(i)$.
4      for $a \in \mathcal{A}$ do
5          Add cut (16b) on $\zeta_a$ using $(x_a^{\mathrm{l}}(i), y_a^{\mathrm{l}}(i))$.
6          Add cut (16h) on $\eta_a$ using $(x_a^{\mathrm{l}}(i), y_a^{\mathrm{l}}(i))$.
7      Set $n^{\mathrm{B}} \leftarrow n^{\mathrm{B}} + 1$, $n^{\mathrm{Z}} \leftarrow n^{\mathrm{Z}} + 1$.
8      if $\mathbf{y}^{\mathrm{l}}(i) \neq \mathbf{y}^{\mathrm{l}}(j)$ for all $j \in \{1 \dots i-1\}$ then
9          Solve TAP $\left(\mathbf{y}^{\mathrm{l}}(i)\right)$ to obtain $\mathbf{x}^{\mathrm{f}}(i)$. Set $UB \leftarrow \min\left\{UB, Z\left(\mathbf{x}^{\mathrm{f}}, \mathbf{y}^{\mathrm{l}}\right)\right\}$.
10         Add outer approximation cut (16g) of $\Delta B$ using $(\mathbf{x}^{\mathrm{f}}(i), \mathbf{y}^{\mathrm{l}}(i))$. Set
            $n^{\mathrm{vf}} \leftarrow n^{\mathrm{vf}} + 1$.
11     Set $i \leftarrow i + 1$.
```

$$Z_{\mathrm{lb}} = \min \quad \sum_{a \in \mathcal{A}} \zeta_a \tag{16a}$$

$$\begin{aligned}
\text{s.t.} \quad \zeta_a \geq\ & x_a^{\mathrm{l}}(i) t_a\left(x_a^{\mathrm{l}}(i), y_a^{\mathrm{l}}(i)\right) + g_a\left(y_a^{\mathrm{l}}(i)\right) \\
& + \left(x_a - x_a^{\mathrm{l}}(i)\right)\left[x_a^{\mathrm{l}}(i)\frac{dt\left(x_a^{\mathrm{l}}(i), y_a^{\mathrm{l}}(i)\right)}{dx_a} + t_a\left(x_a^{\mathrm{l}}(i), y_a^{\mathrm{l}}(i)\right)\right] \\
& + \left(y_a - y_a^{\mathrm{l}}(i)\right)\left[x_a^{\mathrm{l}}(i)\frac{dt\left(x_a^{\mathrm{l}}(i), y_a^{\mathrm{l}}(i)\right)}{dy_a} + \frac{dg\left(y_a^{\mathrm{l}}(i)\right)}{dy_a}\right] \qquad \forall a \in \mathcal{A}, \forall i \in 1 \dots n^{\mathrm{Z}}
\end{aligned} \tag{16b}$$

$$0 \leq y_a \leq \overline{y}_a \qquad \forall a \in \mathcal{A} \tag{16c}$$

$$x_a = \sum_{\pi \in \Pi} \delta_a^{\pi} h^{\pi} \qquad \forall a \in \mathcal{A} \tag{16d}$$

$$\sum_{\pi \in \Pi_{rs}} h^{\pi} = d_{rs} \qquad \forall (r,s) \in \mathcal{Z}^2 \tag{16e}$$

$$h^{\pi} \geq 0 \qquad \forall \pi \in \Pi \tag{16f}$$

$$\begin{aligned}
& \sum_{a \in \mathcal{A}} \eta_a - B(\mathbf{x}^{\mathrm{f}}(i), \mathbf{y}^{\mathrm{l}}(i)) \\
& \quad - \sum_{a \in \mathcal{A}} \left(y_a - y_a^{\mathrm{l}}(i)\right) \int_0^{x_a^{\mathrm{f}}(i)} \frac{\partial t_a\left(\omega, y_a^{\mathrm{l}}(i)\right)}{\partial y_a} d\omega \leq 0 \qquad \forall i \in 1 \dots n^{\mathrm{vf}}
\end{aligned} \tag{16g}$$

$$\begin{aligned}
\eta_a \geq\ & \int_0^{x_a^{\mathrm{l}}} t_a\left(\omega, y_a^{\mathrm{l}}(i)\right) d\omega \\
& + \left(x_a - x_a^{\mathrm{l}}(i)\right) t_a\left(x_a^{\mathrm{l}}(i), y_a^{\mathrm{l}}(i)\right) \\
& + \left(y_a - y_a^{\mathrm{l}}(i)\right) \int_0^{x_a^{\mathrm{l}}(i)} \frac{dt_a\left(\omega, y_a^{\mathrm{l}}(i)\right)}{dy_a} d\omega \qquad \forall a \in \mathcal{A}, \forall i \in 1 \dots n^{\mathrm{B}}
\end{aligned} \tag{16h}$$

To solve CNDP, we propose a cutting-plane algorithm that repeatedly solves LP (16) and gradually adds outer approximation cuts. The pseudo-code of our approach is summarized in Algorithm 1. To avoid path enumeration, we adoption the column generation scheme introduced by Rey & Levin (2024).

3 Numerical results

The main benefit the outer approximation-based cutting-plane algorithm has over published work is the potential to solve CNDP on larger networks than the 6-node, 16-link Harker & Friesz (1984) test network, which was the largest network that other algorithms could be demonstrated on (D. Z. Wang & Lo, 2010; Li et al., 2012; Du & Wang, 2016). We report exact solutions on networks with up to 914 links, which has not previously been achieved in the literature on CNDP.

To demonstrate that our method robustly solves CNDP for these networks, we consider different instances, shown in the "Inst." column of Tables 1 and 2, with different randomly generated costs and different links where $y_a \geq 0$ is permitted. We vary the number of y_a variables that are permitted to be non-zero. As the number of non-zero y_a variables increases, the congestion in the network generally decreases, so instances where $y_a > 0$ is permitted for all links become less interesting. For example, compare the total system travel times in Table 1 for instances on the same network with different numbers of non-zero y_a variables.

Costs per capacity increase are randomly generated because they are typically not published with standard traffic assignment network data unless they are being used for CNDP. y_a was limited to $C_a/2$, for a maximum of a 50% increase in link capacity. (The average value of the link cost is reported in the "avg. cost" column in units of \$ per additional veh/hr capacity. We also varied the number of y_a variables that could be non-negative, shown in the "y_a vars" column of Tables 1 and 2, the average link cost, and the demand.

For each instance, we report the objective, gap at termination, and total system travel time (which excludes the cost of link capacity additions from the objective). The difference between the objective $Z(\mathbf{x}, \mathbf{y})$ and total system travel time indicates when $\mathbf{y} \geq \mathbf{0}$ in the optimal solution. We also show the total system travel time with 0 additional capacity for comparison. The objective and total system travel time are from the best upper bound, i.e. the best `TAP` solution for the $\mathbf{y}$ obtained from problem (16), so they are guaranteed to be follower-optimal for CNDP. We terminated when the gap was below 1%. We also report the total computation time ("Tot. time"), time spent on solving the traffic assignment subproblem ("`TAP` time"), and the number of iterations ("iter.") of Algorithm 1 before termination.

4 Conclusions

CNDP has been extensively studied in the literature, but existing methods for finding exact solutions to CNDP require solving difficult problems creating large computation times for small test networks. Consequently, there was a potential gap to create a method that relies on relatively simple subproblems — linear programs and traffic assignment. Using value function cuts and outer approximation, we created one such algorithm that sequentially solves a linear program and traffic assignment. Convergence occurs because when $\mathbf{x}$ from the high-point relaxation SO-CNDP is not follower-optimal, we add a value function cut based on the follower objective value to exclude $\mathbf{x}$ and other points like it from the SO-CNDP mathematical program. Because such cuts are valid at follower optimality, the revised SO-CNDP mathematical program still finds a lower bound to CNDP, but that lower bound becomes progressively tighter. We used outer approximation to reformulate the nonlinear leader objective and value function cuts into a sequence of linear programs, which is valid when these nonlinear functions are convex (which they are for the Bureau of Public Roads travel time function). Separating the outer approximation by link created a much stronger outer approximation for faster convergence. Column generation was used to more quickly solve the network-based linear programming subproblem.

Due to the computational complexity of their algorithms, prior work on finding exact solutions to CNDP required large computation times to solve CNDP on the small Harker & Friesz (1984) test network with 6 nodes and 16 links. In contrast, we were able to obtain solutions within 1% of global optimality on networks with up to 416 nodes and 914 links in reasonable computation times.

The ability to solve CNDP on much larger networks could be useful for a variety of research and practical problems. Furthermore, we believe that our algorithm is significantly easier to implement than methods from prior work as it relies on linear programs and traffic assignment.

References

Arbex, R. O., & da Cunha, C. B. (2015). Efficient transit network design and frequencies setting multi-objective optimization by alternating objective genetic algorithm. *Transportation Research Part B: Methodological*, *81*, 355–376.

Bar-Gera, H. (2010). Traffic assignment by paired alternative segments. *Transportation Research Part B: Methodological*, *44*(8-9), 1022–1046.

Beckmann, M., McGuire, C. B., & Winsten, C. B. (1956). *Studies in the economics of transportation* (Tech. Rep.).

Dial, R. B. (2006). A path-based user-equilibrium traffic assignment algorithm that obviates path storage and enumeration. *Transportation Research Part B: Methodological*, *40*(10), 917–936.

Du, B., & Wang, D. Z. (2016). Solving continuous network design problem with generalized geometric programming approach. *Transportation Research Record*, *2567*(1), 38–46.

Duran, M. A., & Grossmann, I. E. (1986). An outer-approximation algorithm for a class of mixed-integer nonlinear programs. *Mathematical Programming*, *36*(3), 307–339.

Farahani, R. Z., Miandoabchi, E., Szeto, W. Y., & 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. (2013). A branch and bound algorithm for bi-level discrete network design problem. *Networks and Spatial Economics*, *13*(1), 67–106.

Gairing, M., Harks, T., & Klimm, M. (2017). Complexity and approximation of the continuous network design problem. *SIAM Journal on Optimization*, *27*(3), 1554–1582.

Harker, P. T., & Friesz, T. L. (1984). Bounding the solution of the continous equilibrium network design problem. In *Papers presented during the ninth international symposium on transportation and traffic theory held in delft the netherlands, 11-13 july 1984.*

Leblanc, L. J. (1975). An algorithm for the discrete network design problem. *Transportation Science*, *9*(3), 183–199.

Levin, M. W., Wong, E., Nault-Maurer, B., & Khani, A. (2020). Parking infrastructure design for repositioning autonomous vehicles. *Transportation Research Part C: Emerging Technologies*, *120*, 102838.

Li, C., Yang, H., Zhu, D., & Meng, Q. (2012). A global optimization method for continuous network design problems. *Transportation Research Part B: Methodological*, *46*(9), 1144–1158.

Liu, H., & Wang, D. Z. (2015). Global optimization method for network design problem with stochastic user equilibrium. *Transportation Research Part B: Methodological*, *72*, 20–39.

Nayeem, M. A., Rahman, M. K., & Rahman, M. S. (2014). Transit network design by genetic algorithm with elitism. *Transportation Research Part C: Emerging Technologies*, *46*, 30–45.

Rey, D., & Levin, M. W. (2024). A branch-and-price-and-cut algorithm for discrete network design problems under traffic equilibrium. *Optimization online*. Retrieved from `https://optimization-online.org/?p=28420`

Wang, D. Z., & Lo, H. K. (2010). Global optimum of the linearized network design problem with equilibrium flows. *Transportation Research Part B: Methodological*, *44*(4), 482–492.

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/.*

Table 1: Demonstration of outer approximation algorithm with column generation on larger networks

Network	Non-zero y_a vars	Inst.	Avg. cost	Obj.	gap	TSTT	Tot. time	`TAP` time	iter.
Sioux Falls	10	1	2.39e-01	6959.2	0.94%	6950.4	4.67s	3.77s	9
		2	2.85e-01	6988.3	0.89%	6977.2	7.91s	7.07s	9
24 nodes		3	2.13e-01	6831.0	0.67%	6822.8	5.67s	4.80s	10
76 links		1	2.39e+01	7329.5	0.29%	7201.4	5.77s	4.82s	11
24 zones		2	2.85e+01	7276.5	0.72%	7192.3	5.90s	5.00s	10
		3	2.13e+01	7155.4	0.35%	7031.9	5.19s	4.28s	10
360,600 trips	30	1	2.74e-01	6087.5	0.90%	6060.4	6.94s	5.83s	10
		2	2.93e-01	5909.3	0.93%	5875.1	4.45s	3.58s	9
		3	2.75e-01	5837.0	0.74%	5806.5	5.33s	4.35s	10
		1	2.74e+01	7065.7	0.52%	6691.9	5.75s	4.76s	12
		2	2.93e+01	7079.2	0.56%	6771.8	6.33s	5.35s	12
		3	2.75e+01	7025.4	0.47%	6593.4	6.19s	5.06s	12
	0			7475.7		7475.7			
Eastern Massachusetts	10	1	9.22e-03	2061.6	0.74%	2061.5	99.02s	73.87s	12
		2	7.94e-03	1750.7	0.96%	1750.7	156.99s	103.19s	12
74 nodes		3	9.29e-03	1886.4	0.99%	1886.2	142.77s	124.02s	12
248 links		1	9.22e-01	2063.3	0.71%	2062.7	103.12s	69.50s	12
74 zones		2	7.94e-01	1757.4	0.99%	1752.4	80.59s	58.36s	12
		3	9.29e-01	1892.0	0.90%	1886.0	92.01s	68.52s	11
262,306 trips	30	1	9.47e-03	1287.0	0.96%	1286.8	75.38s	56.85s	13
		2	9.89e-03	1607.0	0.88%	1606.8	85.41s	66.18s	13
		3	9.36e-03	1910.9	0.98%	1910.5	65.19s	44.68s	11
		1	9.47e-01	1303.4	0.97%	1291.1	84.94s	56.97s	12
		2	9.89e-01	1617.8	0.97%	1613.4	128.23s	102.24s	13
		3	9.36e-01	1927.3	0.83%	1918.7	63.44s	41.48s	12
	0			2063.7		2063.7			

Table 2: Demonstration of outer approximation algorithm with column generation on larger networks

Network	Non-zero y_a vars	Inst.	Avg. cost	Obj.	gap	TSTT	Tot. time	`TAP` time	iter.
Berlin Mitte Center	30	1	1.47e-01	12330.2	0.99%	12328.9	599.15s	367.19s	10
		2	1.41e-01	11935.4	0.75%	11933.7	1357.30s	388.01s	11
		3	1.70e-01	12143.6	0.65%	12141.9	1190.32s	281.67s	11
		1	1.47e+00	12338.4	0.98%	12330.9	672.10s	264.45s	10
		2	1.41e+00	11948.7	0.75%	11933.8	953.92s	441.00s	11
		3	1.70e+00	12156.7	0.61%	12142.4	1188.36s	330.88s	11
	60	1	1.40e-01	12107.1	0.73%	12104.3	995.28s	280.07s	11
		2	1.45e-01	12050.4	0.69%	12048.6	997.06s	183.12s	11
		3	1.38e-01	11870.5	0.86%	11866.9	939.05s	344.88s	10
		1	1.40e+00	12127.2	0.97%	12106.5	945.85s	256.55s	10
		2	1.45e+00	12064.2	0.76%	12049.7	1189.48s	214.29s	11
		3	1.38e+00	11895.8	0.98%	11873.2	1131.25s	386.20s	10
	0			12454.1		12454.1			
Anaheim	30	1	2.18e-01	101759.4	0.84%	3574.26	1817.44s	207.73s	6
		2	2.52e-01	101311.5	0.87%	3548.91	1177.66s	190.55s	6
416 nodes		3	2.77e-01	101359.3	0.96%	3559.22	800.60s	253.73s	6
914 links		1	2.18e+01	101845.7	0.94%	3582.84	1811.69s	192.36s	6
38 zones		2	2.52e+01	101738.2	0.95%	3571.52	1094.11s	184.49s	6
		3	2.77e+01	101735.2	0.95%	3575.35	1009.67s	186.35s	6
418,778 trips	60	1	2.28e-01	95760.4	0.97%	3422.53	1951.79s	166.28s	6
		2	2.28e-01	100251.7	0.85%	3507.57	1452.14s	146.00s	6
		3	2.38e-01	100671.8	0.81%	3508.08	1759.30s	207.75s	6
		1	2.28e+01	96193.8	0.95%	3439.27	1438.37s	214.95s	6
		2	2.28e+01	100872.8	0.84%	3529.64	1047.13s	154.29s	6
		3	2.38e+01	101236.6	0.95%	3533.04	2001.19s	240.15s	6
	0			101889.2		101889.2			