



A surrogate-based optimization algorithm for network design problems*

Meng LI¹, Xi LIN¹, Xi-qun CHEN^{†‡2}

⁽¹⁾Department of Civil Engineering, Tsinghua University, Beijing 100084, China)

⁽²⁾College of Civil Engineering and Architecture, Zhejiang University, Hangzhou 310058, China)

[†]E-mail: chenxiqun@zju.edu.cn

Received July 8, 2016; Revision accepted Oct. 27, 2016; Crosschecked Nov. 22, 2017

Abstract: Network design problems (NDPs) have long been regarded as one of the most challenging problems in the field of transportation planning due to the intrinsic non-convexity of their bi-level programming form. Furthermore, a mixture of continuous/discrete decision variables makes the mixed network design problem (MNDP) more complicated and difficult to solve. We adopt a surrogate-based optimization (SBO) framework to solve three featured categories of NDPs (continuous, discrete, and mixed-integer). We prove that the method is asymptotically completely convergent when solving continuous NDPs, guaranteeing a global optimum with probability one through an indefinitely long run. To demonstrate the practical performance of the proposed framework, numerical examples are provided to compare SBO with some existing solving algorithms and other heuristics in the literature for NDP. The results show that SBO is one of the best algorithms in terms of both accuracy and efficiency, and it is efficient for solving large-scale problems with more than 20 decision variables. The SBO approach presented in this paper is a general algorithm of solving other optimization problems in the transportation field.

Key words: Network design problem; Surrogate-based optimization; Transportation planning; Heuristics
<https://doi.org/10.1631/FITEE.1601403>

CLC number: U491; TP202

1 Introduction

The rapid and worldwide growth in population has necessitated building numerous roads to satisfy the increasing traffic demands. It is natural and crucial that decision-makers continuously try to maximize the system performance by building new links or expanding existing ones. Network design problems (NDPs) determine how to allocate a limited budget to

the expansion of capacity for the existing links or the construction of new ones to maximize social welfare or minimize general travel costs. Generally, NDPs can be categorized into three classes: continuous NDPs (CNDPs) where decision variables are continuous, discrete network design problems (DNDPs) where decision variables are discrete (usually binary), and mixed network design problems (MNDPs) which contain both types of decision variables.

NDPs can be formulated as a bi-level programming problem (BLPP), or more specifically, mathematical programming with equilibrium constraints (MPEC) (Luathep *et al.*, 2011). In the BLPP formulation, the upper-level program seeks to optimize the network performance index within the budget, and the lower-level program is the user equilibrium (UE) problem (e.g., Yang and Bell, 1998; Meng *et al.*, 2001) or stochastic user equilibrium (SUE) problem (Davis, 1994). Furthermore, some expansions have

[‡] Corresponding author

* Project supported by the Zhejiang Provincial Natural Science Foundation of China (No. LR17E080002), the National Natural Science Foundation of China (Nos. 51508505, 71771198, 51338008, and 51378298), the Fundamental Research Funds for the Central Universities, China (No. 2017QNA4025), and the Key Research and Development Program of Zhejiang Province, China (No. 2018C01007)

ORCID: Xi-qun CHEN, <http://orcid.org/0000-0001-8285-084X>

© Zhejiang University and Springer-Verlag GmbH Germany 2017

been made to incorporate the uncertainty of travel demand (Yin *et al.*, 2009) and time-dependent (Lo and Szeto, 2009) effects.

Given the intrinsic non-convexity of the problem and the nonlinearity of its objective function, NDPs have been viewed as one of the most challenging problems in transportation. Methods for solving NDP will be reviewed in Sections 1.1–1.3.

1.1 Algorithms for solving continuous network design problems

CNDP has received the greatest attention in transportation literature. Generally, traditional approaches for solving CNDP can be categorized into two major groups: gradient-based algorithms and derivative-free algorithms.

Gradient-based algorithms generally use the sensitivity information of the lower-level UE programming to search local solutions (Meng *et al.*, 2001). Sensitivity analysis-based methods are often incorporated into the optimization procedure of the upper-level objective function (Yang and Yagar, 1995). Chiou (2005) developed a series of methods based on the gradient analysis scheme. These methods have been reported to be efficient for finding a local solution; however, a satisfactory solution quality could not be guaranteed as exhibited by subsequent work.

Derivative-free algorithms include iterative optimization-assignment procedures (Allsop, 1974), Hooke-Jeeves algorithms (Abdulaal and LeBlanc, 1979), equilibrium decomposed optimization (Suwansirikul *et al.*, 1987), simulated annealing (Friesz *et al.*, 1992), and genetic algorithm-based approaches (Yin, 2000). These derivative-free algorithms are incapable of guaranteeing convergence, though it has been reported that some of them can find a good solution after a lengthy computation.

Recently, a mixed-integer linear programming (MILP) approximation approach was introduced to solve CNDP. Wang and Lo (2010) proposed a path-based MILP to solve CNDP and it guarantees a global convergence. Luathep *et al.* (2011) further extended the approach to a link-based MILP formulation to reduce its computational effort while maintaining the convergence property. Liu and Wang (2015) further proposed a linearized technique to solve CNDP with a stochastic user equilibrium. All of

these methods obtained good convergence results on some widely used test networks; however, the computational efforts required by these methods were not that appealing compared to their reported theoretical perfection; i.e., none of those algorithms were tested on networks with more than 20 decision variables.

Thus, this challenge inspired us to develop a solution framework incorporating properties that were good from both theoretical and computational perspectives.

1.2 Algorithms for solving discrete network design problems

LeBlanc (1975) first developed DNDP and proposed a branch-and-bound algorithm to efficiently solve it. The algorithm was tested on the Sioux Falls network with only five binary variables. Poorzahedy and Turnquist (1982) further developed the branch-and-bound algorithm to solve DNDP by approximating the bi-level program through a single-level program; however, the accuracy of the approximation was questioned by other researchers (e.g., Gao *et al.*, 2005). Gao *et al.* (2005) then proposed a generalized Benders' decomposition algorithm to solve DNDP. The algorithm translates the original bi-level problem into a series of master problems and subproblems using a supporting function, and solves some small-scale problems efficiently.

Recently, as in CNDP, an MILP approximation approach has also been introduced to solve DNDP globally. Wang *et al.* (2013) proposed two techniques, i.e., system-optimum (SO) relaxation and UE-reduction, to solve DNDP. The two techniques were solved by using a MILP approach globally. Wang *et al.* (2015) further proposed a linearized approach, which considers optimal new link additions and their optimal capacities simultaneously, to solve DNDP. Farvaresh and Sepehri (2011) also proposed a single-level MI programming approach to solve DNDP.

Many heuristics have also been developed to solve DNDP. This type of algorithm includes genetic algorithms (Cantarella and Vietta, 2006), ant system heuristics (Poorzahedy and Abulghasemi, 2005), and hybrid meta-heuristics (Poorzahedy and Rouhani, 2007).

A major challenge for solving DNDPs nowadays lies in searching for an efficient way to deal with large-scale problems, i.e., problems with more

than 20 decision variables. Most of the algorithms mentioned above were tested on problems with no more than 10 decision variables in their original papers.

1.3 Algorithms for solving mixed network design problems

Compared with CNDP and DNDP, MNDP draws much less attention in the literature. To our best knowledge, the only well-formulated algorithm to solve MNDP is the LMILP proposed by Luatthep *et al.* (2011), where a numerical example was presented to illustrate its solution properties for MNDP on a problem with 10 continuous variables and 10 discrete variables. Other than that, some heuristics (e.g., a scatter search (Gallo *et al.*, 2010)), were proposed in the field.

To make some modest contribution to the rapidly developing NDP research field, we propose a surrogate-based optimization (SBO) method to solve NDP. As shown in the rest of the paper, the proposed framework presents two major advantages compared with most of the following existing methods:

1. Theoretically, while solving CNDP, SBO can guarantee asymptotically complete convergence.

2. Practically, the proposed SBO framework is able to solve a large-scale problem with satisfactory solution qualities for all of these kinds of problems as mentioned above, i.e., CNDP, DNDP, and MNDP.

In recent years, SBO has been applied to transportation research (Chen *et al.*, 2014a; 2014b; 2015a; 2015b; Chow and Regan, 2014). Specifically, Chen *et al.* (2015b) applied an SBO framework to solve NDPs under a dynamic traffic assignment paradigm. Chow and Regan (2014) studied a probabilistic convergent SBO for solving multi-objective robust road pricing, a problem similar to CNDP. However, less work has been carried out in comparing the efficiency and solution quality of SBO with those of existing NDP algorithms.

SBO can be tracked back to the work of efficient global optimization (EGO) proposed by Jones *et al.* (1998). An expected improvement (EI) scheme was proposed to balance local and global searches. However, the original EGO takes a long time to optimize the response surface, and the global convergence is not guaranteed.

To improve the theoretical properties of SBO,

Regis and Shoemaker (2007) proposed a stochastic response surface (SRS) framework for global optimization of the expensive functions. In their work, a candidate point scheme was used as a stochastic iterative process and incorporated to prove that the SBO framework was asymptotically completely convergent. Müller *et al.* (2013) further extended the SRS framework to an SO for a mixed integer (SO-MI) problem framework, which could be used in discrete and MI optimization.

This study combines the merits of both EGO and SRS to propose a framework based on projected candidate point heuristics using the Kriging approximation and EI as the infill criterion. In such a framework, the asymptotically complete convergence is guaranteed when solving continuous problems, and the computational effort will be small for each iteration. This framework is then applied to solve CNDP, DNDP, and MNDP. Numerical experiments are conducted to illustrate the performance of the proposed method. The results show that SBO is among the most advanced algorithms for solving NDPs.

2 Methodology

2.1 Problem formulation

The original NDP can be written as a bi-level program given by (notations are shown in Table 1)

$$\min_{x,y,U} Z_u = \sum_{a \in A} v_a t_a(v_a) + \sum_{a \in A_1} v_a t_a(v_a) + \sum_{a \in A_2} v_a t_a(v_a) \quad (1a)$$

$$\text{s.t.} \quad \sum_{a \in A_1} c_a(y_a) + \sum_{a \in A_2} d_a U_a \leq \text{Budget}, \quad (1b)$$

$$y_a \leq \bar{y}_a, \quad \forall a \in A_1, \quad (1c)$$

$$v_a \leq U_a M, \quad \forall a \in A_2, \quad (1d)$$

$$U_a \in \{0,1\}, \quad \forall a \in A_2, \quad (1e)$$

$$\min_x z_l = \sum_a \int_0^{v_a} t_a(\omega, y_a) d\omega \quad (1f)$$

$$\text{s.t.} \quad \sum_{l \in L^{rs}} f_l^{rs} = q^{rs}, \quad \forall rs \in W, \quad (1g)$$

$$v_a = \sum_{rs} \sum_{l \in L^{rs}} f_l^{rs} \delta_{a,l}^{rs}, \quad \forall rs \in W, \quad a \in A \cup A_1 \cup A_2, \quad (1h)$$

$$f_l^{rs} \geq 0, \quad \forall rs \in W, \quad l \in L^{rs}. \quad (1i)$$

Table 1 Notations

Notation	Explanation
A	The set of unchanged links in the network
A_1	The set of project links for improvement in the network
A_2	The set of new project links in the network
b_i	The minimum of set Θ_i
$c_a(\cdot)$	Monetary cost function of link a
d_a	The cost to build a new link a
F_i	The set of objective function values obtained by Ω_i
f_l^{rs}	Path flow on route l between OD pair rs
\mathbf{G}	Implicit function form of the lower-level UE problem
$\hat{\mathbf{G}}_i(\mathbf{y}, \mathbf{U})$	Kriging approximation of the lower-level UE in iteration i
L^{rs}	The set of routes between OD pair rs
M	An arbitrarily large positive number
n_0	Number of initially evaluated points
q^{rs}	Total demand between OD pair rs
R	The set of origins
S	The set of destinations
$\hat{s}_i^2(\mathbf{y}, \mathbf{U})$	Mean square error of the Kriging approximation in iteration i
t_a	Link travel time function w.r.t. link flow and capacity expansion
U_a	A binary scalar representing the decision whether to build candidate link a
\mathbf{U}	Vector form of U_a
W	The set of OD pairs
v_a	Flow on link a
\mathbf{v}	Vector form of v_a
y_a	The value of capacity expansion on link a
\mathbf{y}	Vector form of y_a
\underline{y}_a	Lower bound of the capacity enhancement of link a
\bar{y}_a	Upper bound of the capacity enhancement of link a
$\delta_{a,l}^{rs}$	1, if route l of rs uses link a ; 0, otherwise
Ω_i	The set of evaluation points in iteration i
Ψ_i	The set of inputs for the Kriging approximation in iteration i (as data points)
Θ_i	The set of inputs for the Kriging approximation in iteration i (as function values)

For upper-level optimizations (1a)–(1e), Z_u of Eq. (1a) is the objective function which is intended to minimize the total travel cost. Note that when the problem is a CNDP, A_2 is an empty set, and when the problem is a DNDP, A_1 is an empty set. Constraint (1b) is the construction budget constraint. Constraint (1c) defines the bounds of the continuous capacity expansion of link a . Constraint (1d) states that only candidate links with $U_a > 0$ are allowed to load a positive link flow, and Eq. (1e) states that U_a is a binary variable.

For the lower-level programming (1f)–(1i), constraint (1g) is the demand conservation condition. The sum of any path flow connecting OD pair rs should be equal to demand q^{rs} . Constraint (1h) is the link-path flow conservation condition. Constraint (1i) limits all path flows to be nonnegative. The Karush–Kuhn–Tucker (KKT) conditions for problem (1f)–(1i) are equivalent to Wardrop’s first principle (Wardrop, 1952).

Problem (1) is a nonlinear and non-convex BLPP. In general, BLPP is non-deterministic polynomial (NP) hard and very difficult to solve. The nonlinearity and non-convexity further increase the difficulty of problem (1). To efficiently solve the above BLPP, we view the lower-level UE problem as an implicit equality constraint of the upper-level problem, and reformulate problem (1) as a one-level optimization problem, given by

$$\min_{\mathbf{y}, \mathbf{U}} Z_u = \sum_{a \in A} v_a t_a(v_a) + \sum_{a \in A_1} v_a t_a(v_a) + \sum_{a \in A_2} v_a t_a(v_a) \quad (2a)$$

$$\text{s.t. constraints (1b)–(1e),} \quad (2b)$$

$$\mathbf{v} = \mathbf{G}(\mathbf{y}, \mathbf{U}), \quad (2c)$$

where objective function (2a) is the same as Eq. (1a), the constraints are the same as those in the upper-level problem (1), and Eq. (2c) is an implicit form of the lower-level UE problem, where \mathbf{y} is the vector of the continuous capacity expansion, \mathbf{U} is the binary vector for the construction of new links, and $\mathbf{G}(\mathbf{y}, \mathbf{U})$ is a mathematically difficult-to-solve function. Though it is too complicated to formulate analytically, some properties (e.g., the uniqueness of \mathbf{v} given \mathbf{y} and \mathbf{U} and the continuity of $\mathbf{G}(\mathbf{y}, \mathbf{U})$ w.r.t. \mathbf{y}) can be proven.

Proposition 1 Given \mathbf{y} and \mathbf{U} , $\mathbf{v} = \mathbf{G}(\mathbf{y}, \mathbf{U})$ is unique if the link travel time function is monotonically increasing.

Proof According to the definition of $\mathbf{G}(\mathbf{y}, \mathbf{U})$, \mathbf{v} is the solution of program (1f)–(1i). If the link travel time function is monotonically increasing, program (1f)–(1i) is convex. Obviously, the solution of a convex program is unique.

Proposition 2 $\mathbf{G}(\mathbf{y}, \mathbf{U})$ is continuous w.r.t. \mathbf{y} if the link travel time function is continuously differentiable and monotonically increasing.

Proof If \mathbf{v}^* solves program (1f)–(1i), then \mathbf{v}^* also solves the following variational inequality problem:

$$\sum_{a \in A \cup A_1 \cup A_2} t_a(v_a^*, y_a)(v_a - v_a^*) \geq 0,$$

$\forall v$ satisfying constraints (1g)–(1i).

Since the polyhedron defined by constraints (1g)–(1i) is a convex and closed set, and the travel time function is assumed to be continuously differentiable, according to the continuity theorem of the parametric variational inequality problem proposed by Harker and Pang (1990), v^* is Lipschitz continuous w.r.t. the variation of y .

Propositions 1 and 2 prove the uniqueness and continuity of $G(y, U)$, which are very important in the proof of an asymptotically complete convergence of SBO. The differentiability of $G(y, U)$ cannot be guaranteed.

2.2 Solution procedure

The description of this framework is shown in Algorithm 1.

Algorithm 1 Surrogate-based optimization for network design problem (SBO-NDP)

- 1 An initial evaluation point set Ω_1 with a Latin hypercube sampling strategy is built. Set $|\Omega_1|=n_0+1$ (where n_0 is the total number of decision variables), $\Psi_0:=\emptyset$, $\Theta_0:=\emptyset$, and iterate number $i=1$.
- 2 Solve static user equilibrium problem (1f)–(1i) on each point in Ω_i , and the corresponding value of objective function (2a) set is F_i . Let $\Psi_i=\Psi_{i-1} \cup \Omega_i$, $\Theta_i=\Theta_{i-1} \cup F_i$.
- 3 Build a Kriging model with inputs Ψ_i and Θ_i . Approximate function $\hat{G}_i(y, U)$ and mean square error (MSE) estimator $\hat{s}_i^2(y, U)$ are obtained.
- 4 If the maximum iteration number has been reached, terminate SBO-NDP; otherwise, solve the following EI maximization problem:

$$\max_{y, U} EI = \frac{1}{\sqrt{2\pi}\hat{s}_i(y, U)} \cdot \int_{-\infty}^{b_i} (b_i - u) \exp\left(-\frac{(u - \hat{G}_i(y, U))^2}{2\hat{s}_i^2(y, U)}\right) du \quad (3)$$

s.t. constraints (1b)–(1e) and $b_i = \min(\Theta_i)$ with projected candidate point heuristics. The solution of problem (3) is (y_i^*, U_i^*) , and the corresponding EI value is I_i .

- 5 (For DNDP only)

- (1) If (y_i^*, U_i^*) coincides with some vectors in Ψ_i , solve optimization problem (3) s.t. constraints (1b)–(1e) and $EI \leq I_i - \varepsilon$, where ε is a sufficiently small positive number. The solution is (y_i^*, U_i^*) , and the corresponding EI value is I_i . Otherwise, go to step 6.
 - (2) If $I_i \leq 0$, then each feasible point in the solution space is in Ψ_i , and terminate SBO-NDP; otherwise, go to step 5(1).
- 6 Set $\Omega_{i+1} = \{(y_i^*, U_i^*)\}$, and iteration number $i=i+1$.
-

The explanations of the SBO-NDP procedure above are briefly listed in Sections 2.2.1–2.2.4.

2.2.1 Latin hypercube sampling strategy

A proper scheme to carefully allocate initial evaluation points is needed to enhance the accuracy of the approximation. The Latin hypercube sampling strategy (Nielsen *et al.*, 2002) is one of the most widely used methods. See Chen *et al.* (2014b) for a detailed discussion.

2.2.2 Kriging model

The Kriging model is a powerful tool for the complicated function evaluation. The core idea is to approximate an unknown function by introducing a Gaussian process. In this model, a linear function with Gaussian white noise is used as a metamodel for the true but unknown function. Parameters can be estimated via a maximum likelihood estimation (MLE). The Kriging model provides not only an approximation result but also MSE at each point in the feasible domain.

2.2.3 Expected improvement

Jones *et al.* (1998) claimed that EI was an attractive way to balance the local and global searches. The objective of EI is given by Eq. (3), which is to maximize the expectation of a given point to become a potential global optimum.

2.2.4 Projected candidate point heuristics

In the stochastic response surface framework, a candidate point strategy is used to find a near-optimal solution based on the surrogate model. Candidate point heuristics generate several groups of candidate points that belong to a normal distribution or uniform

distribution, to capture both local and global properties.

In the SBO-NDP framework, a projected version of candidate point heuristics is used to handle non-box constraints. It projects each candidate point into the feasible region which is assumed to be convex, thus keeping every iteration of the solution procedure from violating constraints. Two aspects should be mentioned for projecting discrete variables or mixed variables without violating convergence:

1. When generating candidate points for DNDP, deleting infeasible ones still guarantees convergence.
2. When generating candidate points for MNDP, discrete variables are checked first. If the construction cost of discrete variables violates the budget constraint, the deletion still guarantees convergence; otherwise, continuous variables are projected, so that the total construction cost is within the limit.

2.3 Proof of convergence

We now prove the asymptotically complete convergence of the SBO-NDP framework. Most existing algorithms for solving NDPs are actually incomplete except some newly developed linear relaxation ones.

Regis and Shoemaker (2007) provided a proof by discussing the convergence properties of SBO for box-constrained problems. In this paper, we extend the convergence result by projecting candidate points into a convex feasible domain; thus, the asymptotically complete convergence of the SBO-NDP framework on any convex set can be proved. However, some lemmas are required before proving the final results. They are put in Appendix because the proof procedure is quite complicated.

Proposition 3 SBO-NDP is an asymptotically complete convergence for CNDP.

Proof According to the continuity of $G(\mathbf{y}, \mathbf{U})$, which is proven in Proposition 2, the proof of the asymptotically complete convergence could be directly obtained from Lemma A2 in Appendix, considering that the feasible set defined by constraints (1b)–(1e) is convex.

For DNDP, we note that step 5 in the SBO-NDP framework (Algorithm 1) is a safeguard to prevent repeated solutions. It is straightforward to conclude that SBO reaches the global optimum within finite

iterations because the cardinality of the feasible set is finite.

For MNDP, the asymptotically complete convergence cannot be guaranteed due to its complexity. However, since the number of combinations of feasible discrete variables is finite, then MNDP can be viewed as a finite number of CNDPs. According to Proposition 3, given an indefinitely long run, the solution given by SBO-NDP is the global optimum.

3 Numerical examples

This section shows the tests of SBO-NDP's performance. The CPU time for the computation of a single UE assignment on the Sioux Falls network is about 0.3–0.4 s. In each scenario, a numerical example with no less than 20 decision variables is presented.

3.1 Continuous network design problem

3.1.1 A 16-link network

The 16-link network is illustrated in Fig. 1. It consists of six nodes and two OD pairs, i.e., (1, 6) and (6, 1). All inputs of this network can be found in Suwansirikul *et al.* (1987). In this study, a demand of 10 is used to illustrate the algorithm performance. The upper-level objective function of CNDP is $Z = \sum_{a \in A \cup A_1} x_a t_a(x_a) + \theta \sum_{a \in A_1} c_a y_a$, which is the sum of the total network travel time and equivalent link construction costs, where θ is the conversion factor, and in this case was set to 1. To cope with the existing results of other algorithms, the budget constraint was transferred to a penalty term in the objective function. There were 16 continuous decision variables. The maximum iteration of SBO was set to 100.

Table 2 shows a comparison of performance among different algorithms. Since SBO-NDP is a stochastic optimization algorithm, to demonstrate its robustness, 20 random runs were conducted. The results of the best, worst, and medium ones are shown in Table 2. Note that the optimal objective function values Z 's of the existing algorithms were recomputed to guarantee consistency. The optimal objective function values reported in the literature are denoted by Z_o , which can be slightly different from Z .

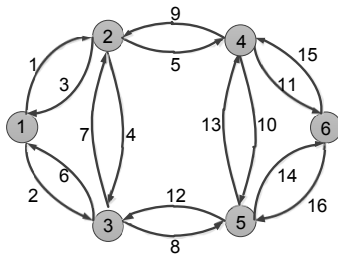


Fig. 1 The 16-link test network

Table 2 Continuous network design problem solving results using different algorithms on a 16-link network

Abbreviation	Algorithm	Source	Z	Z _o
IOA	Iterative optimization assignment	Allsop (1974)	557.84	556.61
HJ	Hooke-Jeeves algorithm	Abdulaal and LeBlanc (1979)	562.15	557.22
EDO	Equilibrium decomposed optimization	Suwansirikul et al. (1987)	540.20	540.74
MINOS	Modular in-core nonlinear system	Suwansirikul et al. (1987)	557.18	557.14
SA	Simulated annealing	Friesz et al. (1992)	526.75	528.50
SAB	Sensitivity analysis based	Yang and Yagar (1995)	536.88	536.08
AL	Augmented Lagrangian	Meng et al. (2001)	532.57	532.71
GP	Gradient projection	Cantarella and Vitetta (2006)	533.08	534.02
CGP	Conjugated gradient projection	Chiou (2005)	535.27	534.11
QNEW	Quasi-Newton	Chiou (2005)	535.64	534.08
PT	PARTAN GP	Chiou (2005)	534.49	534.02
PMILP	Path-based MILP	Wang and Lo (2010)	522.31	523.63
LMILP	Link-based MILP	Luathep et al. (2011)	526.08	526.49
PMC	Penalty with multicutting plane	Li et al. (2012)	522.73	522.74
SBO (best)	Surrogate-based optimization	This paper	521.24	–
SBO (medium)	Surrogate-based optimization	This paper	522.40	–
SBO (worst)	Surrogate-based optimization	This paper	525.42	–

Bold number indicates the best result

In Table 2, the best result among 20 random runs of SBO is 521.24, which is also the best among all the algorithms listed and even better than the results given by PMILP (522.31) and PMC (522.73). The 10th best result for SBO (522.40) is only slightly worse than that of PMILP, and the worst SBO result (525.42) is only worse than those of PMILP and PMC.

As shown in Table 3, since the solutions of SA, PMILP, LMILP, PMC, and SBO are comparable, we briefly compare the computational efficiency. Considering the extremely low computational effort (30 s) and iteration number (100) required to achieve a solution that is both of high quality and robust, we can reasonably state that SBO is a very efficient method for solving CNDP.

Table 3 Computation time comparison for solving for the continuous network design problem

Algorithm	Test computer	Computation effort
SA	–	More than 10 000 UE assignments
PMILP	Dell GX260 3 GB RAM	22.8 h
PMC	Intel core 2 duo, 2.53 GHz CPU, 4 GB RAM	12.0 h
SBO	2.4 GHz intel core, 4 GB RAM	30 s; 100 UE assignments

3.1.2 Sioux Falls network

The Sioux Falls network has been used widely. The basic input can also be found in Suwansirikul et al. (1987). Fig. 2 illustrates the topology of the Sioux Falls network and links to be expanded.

The objective function is $Z = \sum_{a \in A \cup A_1} x_a t_a(x_a) + 0.001$

$\cdot \sum_{a \in A_1} c_a y_a^2$, which is the sum of the total network travel

time and a nonlinear transformation of the link construction costs. To illustrate the SBO-NDP capability, we set 30 continuous decision variables as shown in Fig. 2. As a comparison, two well-known heuristics, i.e., the genetic algorithm (GA) and the simulated annealing (SA), were tested.

Every algorithm was tested 20 times. The best and worst results are presented in Table 4. The maximum number of SBO iterations was set to 200, the maximum number of GA generations was set to 100 with a population of 50 in each generation, and the maximum number of SA iterations was set at 5000.

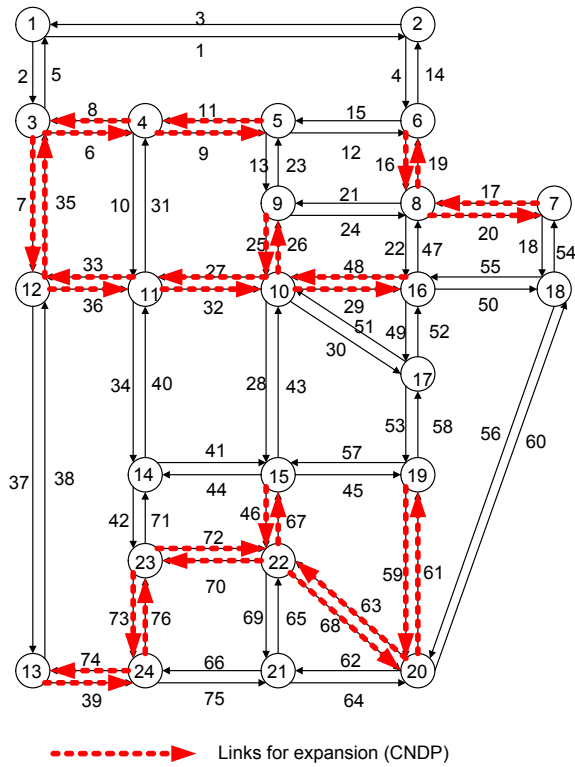


Fig. 2 Sioux Falls network for continuous network design problem

Table 4 Algorithm performance comparison for solving continuous design problems on the Sioux Falls network

Algorithm	Z	Number of UE assignments	Average CPU time (min)
GA (best)	72.98	5000	78
GA (worst)	74.62	5000	78
SA (best)	78.62	5000	75
SA (worst)	81.43	5000	75
SBO (best)	72.62	200	5
SBO (worst)	73.24	200	5

Bold number indicates the best result

In Table 4, it is clear that SBO achieves better solutions than GA and SA. The worst solution for SBO is only 1% worse than the best solution, indicating that SBO can obtain a robust solution within only 200 iterations for 30 decision variables.

3.2 Discrete network design problem

We built a network of 16 nodes, 18 existing links, and 24 candidate links to test the SBO performance for solving DNDP, as shown in Fig. 3. There are two OD pairs, (1, 16) and (4, 13), with demands of 30 and

40, respectively. The maximum iteration number is 200. We ran each algorithm 20 times for each budget level.

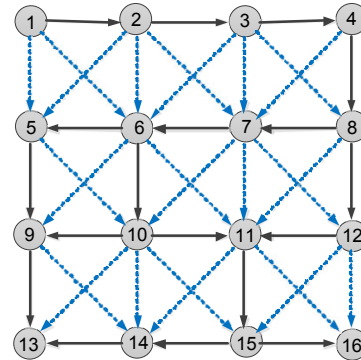


Fig. 3 The 24-candidate-link network

In Table 5, SBO reaches the optimum with a probability of more than 0.6 at each budget level. The results indicate that SBO is able to solve a discrete problem with 24 binary variables efficiently and precisely.

Table 5 Discrete network design problem solving results using different algorithms on a 24-candidate-link network

Budget	Z	Probability of reaching the optimum
3000	120.410	1.00
4000	87.790	0.85
5000	67.813	0.80
6000	66.050	0.65
7000	65.020	0.85

The total cost of all candidate links is 10 850

In Table 6, the budget is set at 6000. The global optimal value is 66.05. Similar to CNDP, SBO obtains the most precise, efficient, and robust solutions.

Table 6 Algorithm performance comparison for solving discrete network design problem on a 24-candidate-link network

Algorithm	Z	Number of UE assignments	Average CPU time (min)
GA (best)	66.05	5000	25
GA (worst)	82.43	5000	25
SA (best)	69.39	5000	23
SA (worst)	84.37	5000	23
SBO (best)	66.05	200	1
SBO (worst)	67.23	200	1

Bold number indicates the global optimal value. Budget=6000

3.3 Mixed network design problem

The test network topology is shown in Fig. 4. Its input can be found in Luathep *et al.* (2011). Table 7 illustrates the comparison results of LMILP and SBO, showing that the best and medium solutions for SBO outperform LMILP in terms of quality, and the worst solution still has some acceptance.

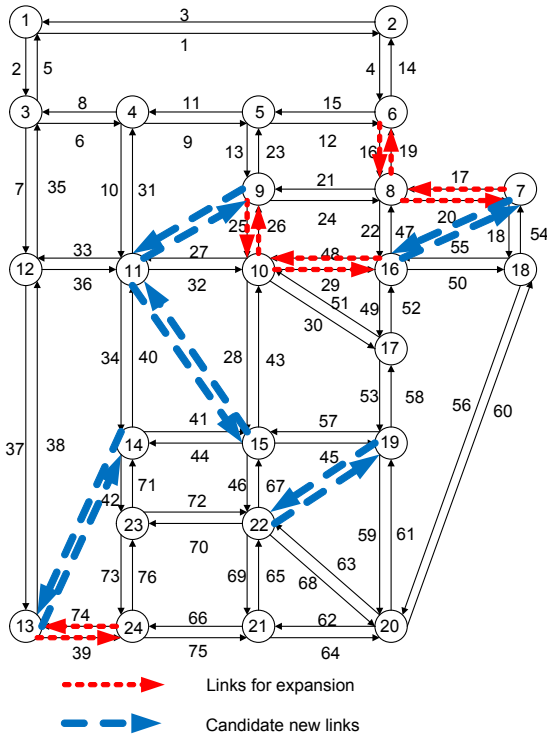


Fig. 4 Sioux Falls network for mixed network design problems

Table 7 Mixed network design problem solving results using different algorithms on the Sioux Falls network

Algorithm	Z
LMILP	67.27
SBO (best)	66.98
SBO (medium)	67.10
SBO (worst)	67.64

Bold number indicates the best results. Budget=4 000 000

Table 8 presents a comparison of algorithm performance for SBO, GA, and SA. The maximum number of generations for GA was set to 50 with a population of 50 in each generation, and the maximum iteration number for SA was set to 2500. We can see that SBO again greatly outperforms GA and SA in terms of both efficiency and quality.

Table 8 Algorithm performance comparison for mixed network design problem on the Sioux Falls network

Algorithm	Z	Number of UE assignments	Average CPU time (min)
GA (best)	67.40	2500	43
GA (worst)	74.12	2500	43
SA (best)	72.19	2500	40
SA (worst)	80.29	2500	40
SBO (best)	66.98	100	3
SBO (worst)	67.64	100	3

Bold number indicates the global optimal value

4 Conclusions

In this paper, we developed a surrogate-based optimization framework to solve NDPs including CNDP, DNDP, and MNDP. SBO is a promising algorithm for solving NDPs for the following reasons:

1. Theoretically, SBO for solving CNDP is asymptotically completely convergent. Given an indefinitely long run time, the algorithm was proven to reach the global optimum with probability one.

2. Practically, SBO works well and outperforms other existing algorithms in different numerical tests. In the problem with 10–20 decision variables, the SBO solution quality is comparable to the best results obtained by existing algorithms but with a relatively low computation cost. In the problem with more than 20 decision variables, SBO performs much better than other heuristics (e.g., GA and SA).

There is still room for future research. First, though guaranteeing asymptotically complete convergence, SBO is still intrinsically heuristic because it cannot estimate the gap between the current solution and the true global optimum. Second, SBO assumes a specific surrogate form, which may be inappropriate when the true function is highly singular.

References

Abdulaal, M., LeBlanc, L.J., 1979. Continuous equilibrium network design models. *Transp. Res. B*, **13**(1):19-32. [https://doi.org/10.1016/0191-2615\(79\)90004-3](https://doi.org/10.1016/0191-2615(79)90004-3)

Allsop, R.E., 1974. Some possibilities for using traffic control to influence trip distribution and route choice. Proc. 6th Int. Symp. on Transportation and Traffic Theory, p. 345-373.

Cantarella, G.E., Vitetta, A., 2006. The multi-criteria road network design problem in an urban area. *Transportation*, **33**(6):567-588. <https://doi.org/10.1007/s11116-006-7908-z>

- Chen, X.Q., Yin, M.G., Song, M.Z., et al., 2014a. Social welfare maximization of multimodal transportation. *Transp. Res. Rec. J. Transp. Res. Board*, **2451**:36-49. <https://doi.org/10.3141/2451-05>
- Chen, X.Q., Zhang, L., He, X., et al., 2014b. Surrogate-based optimization of expensive-to-evaluate objective for optimal highway toll charges in transportation network. *Comput.-Aid. Civil Infrastr. Eng.*, **29**(5):359-381. <https://doi.org/10.1111/mice.12058>
- Chen, X.Q., Zhu, Z., Zhang, L., 2015a. Simulation-based optimization of mixed road pricing policies in a large real-world network. *Transp. Res. Proc.*, **8**:215-226. <https://doi.org/10.1016/j.trpro.2015.06.056>
- Chen, X.Q., Zhu, Z., He, X., et al., 2015b. Surrogate-based optimization for solving a mixed integer network design problem. *Transp. Res. Rec. J. Transp. Res. Board*, **2497**:124-134. <https://doi.org/10.3141/2497-13>
- Chiou, S.W., 2005. Bilevel programming for the continuous transport network design problem. *Transp. Res. B*, **39**(4):361-383. <https://doi.org/10.1016/j.trb.2004.05.001>
- Chow, J.Y.J., Regan, A.C., 2014. A surrogate-based multi-objective metaheuristic and network degradation simulation model for robust toll pricing. *Optim. Eng.*, **15**(1):137-165. <https://doi.org/10.1007/s11081-013-9227-5>
- Davis, G.A., 1994. Exact local solution of the continuous network design problem via stochastic user equilibrium assignment. *Transp. Res. B*, **28**(1):61-75. [https://doi.org/10.1016/0191-2615\(94\)90031-0](https://doi.org/10.1016/0191-2615(94)90031-0)
- Farvaresh, H., Sepehri, M.M., 2011. A single-level mixed integer linear formulation for a bi-level discrete network design problem. *Transp. Res. E*, **47**(5):623-640. <https://doi.org/10.1016/j.tre.2011.02.001>
- Friesz, T.L., Cho, H.J., Mehta, N.J., et al., 1992. A simulated annealing approach to the network design problem with variational inequality constraints. *Transp. Sci.*, **26**(1):18-26. <https://doi.org/10.1287/trsc.26.1.18>
- Gallo, M., D'Acerno, L., Montella, B., 2010. A meta-heuristic approach for solving the urban network design problem. *Eur. J. Oper. Res.*, **201**(1):144-157. <https://doi.org/10.1016/j.ejor.2009.02.026>
- Gao, Z.Y., Wu, J.J., Sun, H.J., 2005. Solution algorithm for the bi-level discrete network design problem. *Transp. Res. B*, **39**(6):479-495. <https://doi.org/10.1016/j.trb.2004.06.004>
- Harker, P.T., Pang, J.S., 1990. Finite-dimensional variational inequality and nonlinear complementarity problems: a survey of theory, algorithms and applications. *Math. Program.*, **48**(1-3):161-220. <https://doi.org/10.1007/BF01582255>
- Jones, D.R., Schonlau, M., Welch, W.J., 1998. Efficient global optimization of expensive black-box functions. *J. Glob. Optim.*, **13**(4):455-492. <https://doi.org/10.1023/A:1008306431147>
- LeBlanc, L.J., 1975. An algorithm for the discrete network design problem. *Transp. Sci.*, **9**(3):183-199. <https://doi.org/10.1287/trsc.9.3.183>
- Li, C.M., Yang, H., Zhu, D.L., et al., 2012. A global optimization method for continuous network design problems. *Transp. Res. B*, **46**(9):1144-1158. <https://doi.org/10.1016/j.trb.2012.05.003>
- Liu, H.X., Wang, D.Z.W., 2015. Global optimization method for network design problem with stochastic user equilibrium. *Transp. Res. B*, **72**:20-39. <https://doi.org/10.1016/j.trb.2014.10.009>
- Lo, H.K., Szeto, W.Y., 2009. Time-dependent transport network design under cost-recovery. *Transp. Res. B*, **43**(1):142-158. <https://doi.org/10.1016/j.trb.2008.06.005>
- Luathep, P., Sumalee, A., Lam, W.H.K., et al., 2011. Global optimization method for mixed transportation network design problem: a mixed-integer linear programming approach. *Transp. Res. B*, **45**(5):808-827. <https://doi.org/10.1016/j.trb.2011.02.002>
- Meng, Q., Yang, H., Bell, M.G.H., 2001. An equivalent continuously differentiable model and a locally convergent algorithm for the continuous network design problem. *Transp. Res. B*, **35**(1):83-105. [https://doi.org/10.1016/S0191-2615\(00\)00016-3](https://doi.org/10.1016/S0191-2615(00)00016-3)
- Müller, J., Shoemaker, C.A., Piché, R., 2013. SO-MI: a surrogate model algorithm for computationally expensive nonlinear mixed-integer black-box global optimization problems. *Comput. Oper. Res.*, **40**(5):1383-1400. <https://doi.org/10.1016/j.cor.2012.08.022>
- Nielsen, H.B., Lophaven, S.N., Søndergaard, J., 2002. DACE—a MATLAB Kriging Toolbox. Technical University of Denmark. Available from <http://www2.imm.dtu.dk/projects/dace> [Accessed on July 8, 2016].
- Poorzahedy, H., Turnquist, M.A., 1982. Approximate algorithms for the discrete network design problem. *Transp. Res. B*, **16**(1):45-55. [https://doi.org/10.1016/0191-2615\(82\)90040-6](https://doi.org/10.1016/0191-2615(82)90040-6)
- Poorzahedy, H., Abulghasemi, F., 2005. Application of ant system to network design problem. *Transportation*, **32**(3):251-273. <https://doi.org/10.1007/s11116-004-8246-7>
- Poorzahedy, H., Rouhani, O.M., 2007. Hybrid meta-heuristic algorithms for solving network design problem. *Eur. J. Oper. Res.*, **182**(2):578-596. <https://doi.org/10.1016/j.ejor.2006.07.038>
- Regis, R.G., Shoemaker, C.A., 2007. A stochastic radial basis function method for the global optimization of expensive functions. *INFORMS J. Comput.*, **19**(4):497-509. <https://doi.org/10.1287/ijoc.1060.0182>
- Suwansirikul, C., Friesz, T.L., Tobin, R.L., 1987. Equilibrium decomposed optimization: a heuristic for the continuous equilibrium network design problem. *Transp. Sci.*, **21**(4):254-263. <https://doi.org/10.1287/trsc.21.4.254>
- Wang, D.Z.W., Lo, H.K., 2010. Global optimum of the linearized network design problem with equilibrium flows. *Transp. Res. B*, **44**(4):482-492. <https://doi.org/10.1016/j.trb.2009.10.003>
- Wang, D.Z.W., Liu, H.X., Szeto, W.Y., 2015. A novel discrete network design problem formulation and its global optimization solution algorithm. *Transp. Res. E*, **79**:213-230. <https://doi.org/10.1016/j.tre.2015.04.005>

Wang, S.A., Meng, Q., Yang, H., 2013. Global optimization methods for the discrete network design problem. *Transp. Res. B*, **50**:42-60.
<https://doi.org/10.1016/j.trb.2013.01.006>

Wardrop, J.G., 1952. Road Paper. Some theoretical aspects of road traffic research. *Proc. Inst. Civil Eng.*, **1**(3):325-362.
<https://doi.org/10.1680/ipeds.1952.11259>

Yang, H., Bell, M.G.H., 1998. Models and algorithms for road network design: a review and some new developments. *Transp. Rev.*, **18**(3):257-278.
<https://doi.org/10.1080/01441649808717016>

Yang, H., Yagar, S., 1995. Traffic assignment and signal control in saturated road networks. *Transp. Res. A*, **29**(2):125-139. [https://doi.org/10.1016/0965-8564\(94\)E0007-V](https://doi.org/10.1016/0965-8564(94)E0007-V)

Yin, Y.F., 2000. Genetic-algorithms-based approach for bilevel programming models. *J. Transp. Eng.*, **126**(2):115-120.

Yin, Y.F., Madanat, S.M., Lu, X.Y., 2009. Robust improvement schemes for road networks under demand uncertainty. *Eur. J. Oper. Res.*, **198**(2):470-479.
<https://doi.org/10.1016/j.ejor.2008.09.008>

Appendix: Proof of asymptotically complete convergence of Algorithm 1

Notations in Appendix are shown in Table A1.

Table A1 Notations for Appendix

Notation	Explanation
$B(x, \delta)$	Neighborhood of x with radius δ
$B_D(x, \delta)$	Intersect of $B(x, \delta)$ and feasible domain D
$B_H(x, \delta)$	Intersect of $B(x, \delta)$ and hypercube H
D	Feasible domain
$e_{n,j}$	j th candidate point vector in iteration n
$E_{n,j}$	Random vector representing $e_{n,j}$
H	Hypercube in which candidate points are sampled
x_n	Function evaluation point at iteration n
X_n	Random vector representing x_n
A_n	Variance of normal distribution in iteration n
\mathfrak{F}_n	Set $\{X_1, X_2, \dots, X_{n_0}, E_{n_0,1}, E_{n_0,2}, \dots, E_{n_0,t}, \dots, E_{n,1}, E_{n,2}, \dots, E_{n,t}\}$
$\sigma(\cdot)$	σ -field

Lemma A1 Let f be a function defined on a convex domain $D \subseteq \mathfrak{R}^d$ and suppose that x^* is the unique global minimizer of continuous function f on D in the sense that $f(x^*) = \inf_{x \in D} f(x) > -\infty$ and $f(x) > \inf_{x \in D, \|x-x^*\| \geq \eta} f(x) > f(x^*)$ for all $\eta > 0$. Define the

sequence of random vectors $\{X_n^*\}_{n \geq 1}$ as follows: $X_1^* = X_1$ and $X_n^* = X_n$ if $f(X_n) < f(X_n^*)$, while $X_n^* = X_{n-1}$ otherwise. If the candidate point strategy in this study satisfies the following Conditions 1 and 2, then $X_n^* \rightarrow x^*$ is almost for sure:

Condition 1 For each $n \geq n_0$, $E_{n,1}, E_{n,2}, \dots, E_{n,t}$ are conditionally independent given the random vectors in \mathfrak{F}_{n-1} .

Condition 2 For any $j=1, 2, \dots, t, x \in D$, and $\delta > 0$, there exists $v_j(x, \delta) > 0$, such that $P\{E_{n,j} \in B(x, \delta) \cap D \mid \sigma(\mathfrak{F}_{n-1})\} \geq v_j(x, \delta)$.

Proof See the proof of Theorem 1 in Regis and Shoemaker (2007).

Lemma A2 The projected candidate point heuristics for problems with a convex feasible domain satisfies Conditions 1 and 2 in Lemma A1.

Proof The two groups of candidate points actually obey the normal distribution and uniform distribution as mentioned in Section 3.2, respectively. Then it is sufficient to prove that the normal distribution and uniform distribution candidate points both satisfy Conditions 1 and 2. It is obvious that Condition 1 holds for both generated groups of points. Thus, we need only to prove that Condition 2 holds.

As defined in Section 2.2, in each iteration, projected candidate point heuristics sample candidate points on a hypercube H , and then project these candidate points into the convex feasible domain $D \subseteq H$. We define

$$B_D(x, \delta) = B(x, \delta) \cap D,$$

$$B_H(x, \delta) = B(x, \delta) \cap H,$$

and $\text{Pro}_D(E_{n,t})$ represents the projection of random vector $E_{n,t}$ on D . It is clear that $B_D(x, \delta) \subseteq B_H(x, \delta)$. We first prove that for any $x \in D$,

$$P\{\text{Pro}_D(E_{n,j}) \in B_D(x, \delta) \mid \sigma(\mathfrak{F}_{n-1})\} \geq P\{E_{n,j} \in B_H(x, \delta) \mid \sigma(\mathfrak{F}_{n-1})\}.$$

To prove the above inequality, we equivalently prove that for any $x \in D$, if $E_{n,j} \in B_H(x, \delta)$, then $\text{Pro}_D(E_{n,j}) \in B_D(x, \delta)$. This leads to the following two scenarios:

1. $\mathbf{E}_{n,j} \in B_D(\mathbf{x}, \delta)$, then $\mathbf{Pro}_D(\mathbf{E}_{n,j}) = \mathbf{E}_{n,j}$. It indicates that $\mathbf{Pro}_D(\mathbf{E}_{n,j}) \in B_D(\mathbf{x}, \delta)$.

2. $\mathbf{E}_{n,j} \in B_H(\mathbf{x}, \delta)$ and $\mathbf{E}_{n,j} \notin B_D(\mathbf{x}, \delta)$, if $\mathbf{Pro}_D(\mathbf{E}_{n,j}) \notin B_D(\mathbf{x}, \delta)$, then $\|\mathbf{x} - \mathbf{E}_{n,j}\| \leq \delta$ and $\|\mathbf{x} - \mathbf{Pro}_D(\mathbf{E}_{n,j})\| > \delta$. Thus, we have

$$\begin{aligned} & (\mathbf{Pro}_D(\mathbf{E}_{n,j}) - \mathbf{E}_{n,j})(\mathbf{Pro}_D(\mathbf{E}_{n,j}) - \mathbf{x}) \\ &= (\mathbf{Pro}_D(\mathbf{E}_{n,j}) - \mathbf{x} - \mathbf{E}_{n,j} + \mathbf{x})(\mathbf{Pro}_D(\mathbf{E}_{n,j}) - \mathbf{x}) \\ &= (\mathbf{Pro}_D(\mathbf{E}_{n,j}) - \mathbf{x})^2 - (\mathbf{E}_{n,j} - \mathbf{x})(\mathbf{Pro}_D(\mathbf{E}_{n,j}) - \mathbf{x}) \\ &\geq (\mathbf{Pro}_D(\mathbf{E}_{n,j}) - \mathbf{x})^2 - \delta \|\mathbf{Pro}_D(\mathbf{E}_{n,j}) - \mathbf{x}\| \\ &> 0. \end{aligned}$$

This contradicts the theorem of projection onto a convex set. Thus, $\mathbf{Pro}_D(\mathbf{E}_{n,j}) \in B_D(\mathbf{x}, \delta)$ must hold. Combined with Scenario 1, we successfully prove the inequality above.

Define $\Psi_H(\delta) := \inf_{\mathbf{x} \in H} \mu(B_H(\mathbf{x}, \delta))$, where μ is the Lebesgue measure on \mathfrak{R}^d . Since H is a compact hypercube, it is straightforward to conclude that $\Psi_H(\delta) > 0$ for any $\delta > 0$ (Regis and Shoemaker, 2007). Then we are ready to show that candidate points obeying both uniform and normal probabilistic distributions satisfy Condition 2.

Uniform distribution: We show that

$$\begin{aligned} & P\{\mathbf{Pro}_D(\mathbf{E}_{n,j}) \in B_D(\mathbf{x}, \delta) | \sigma(\mathfrak{T}_{n-1})\} \\ &\geq P\{\mathbf{E}_{n,j} \in B_H(\mathbf{x}, \delta) | \sigma(\mathfrak{T}_{n-1})\} \\ &= \mu(B_H(\mathbf{x}, \delta)) / \mu(H) \geq \Psi_H(\delta) / \mu(H) > 0. \end{aligned}$$

Thus, Condition 2 holds for the uniform distribution.

Normal distribution: We can easily prove that the conditional distribution of such candidate points on hypercube H satisfies

$$\begin{aligned} h_{n,j}(\mathbf{e} | \sigma(\mathfrak{T}_{n-1})) &= (2\pi A_n^2)^{-d/2} \exp\left\{-\frac{\|\mathbf{e} - \mathbf{x}_n^*\|^2}{2A_n^2}\right\} \\ &\geq \left(2\pi(\sup_{n \geq n_0} A_n)^2\right)^{-d/2} \exp\left\{-\frac{\sup_{\mathbf{x} \in H} \|\mathbf{x} - \mathbf{x}_n^*\|^2}{2(\inf_{n \geq n_0} A_n)^2}\right\} \\ &:= C > 0, \end{aligned}$$

where A_n is the variance of the normal distribution in iteration n . Thus, we have

$$\begin{aligned} & P\{\mathbf{Pro}_D(\mathbf{E}_{n,j}) \in B_D(\mathbf{x}, \delta) | \sigma(\mathfrak{T}_{n-1})\} \\ &\geq P\{\mathbf{E}_{n,j} \in B_H(\mathbf{x}, \delta) | \sigma(\mathfrak{T}_{n-1})\} \\ &= \int_{B_H(\mathbf{x}, \delta)} h_{n,j}(\mathbf{e} | \sigma(\mathfrak{T}_{n-1})) d\mathbf{e} \\ &\geq C \mu(B_H(\mathbf{x}, \delta)) \geq C \Psi_H(\delta) > 0. \end{aligned}$$

Thus, the proof of Lemma 2 is completed.