URAL MATHEMATICAL JOURNAL, Vol. 11, No. 2, 2025, pp. 200–214

DOI: 10.15826/umj.2025.2.014

IMPROVED BRANCH-AND-PRICE ALGORITHM FOR THE EFFICIENT 2-TERMINAL RELIABILITY PROBLEM

Roman A. Rudakov

Krasovskii Institute of Mathematics and Mechanics,
Ural Branch of the Russian Academy of Sciences,
16 S. Kovalevskaya Str., Ekaterinburg, 620108, Russian Federation

r.rudakov@imm.uran.ru

Abstract: The Efficient 2-Terminal Reliability Problem is a nonlinear optimization problem aimed at designing a minimal-cost network within the reliability guaranties. Recent research has provided Branch-and-Price (BnP) solution based on probability relaxation, the Dantzig–Wolfe decomposition, followed by the column generation technique, and Branch-and-Bound scheme. Unfortunately, the performance of this algorithm deteriorates in cases of high-density graphs and stringent unreliability thresholds. By extending our recent approach, we introduce an improved BnP algorithm supplemented with novel valid inequalities, more efficient nonlinear integer pricing problem solver, primal heuristics, and branching strategies. Evaluation results on benchmarking instances demonstrate significant performance advantage of the proposed method.

Keywords: 2-Terminal Reliable Problem, Dantzig–Wolfe decomposition, Branch-and-price.

1. Introduction

Recent global events demonstrate the vulnerability of modern business operations [3]. Political, natural, or industry-related incidents can disrupt logistics and IT infrastructures, restrict the supply of critical resources, and lead to energy shortages. Therefore, ensuring reliable supply chains and network robustness is essential.

The Erdös–Rényi random graph model is among the simplest and most widely used approaches for analyzing potential network failures. In such networks, the probability of a route existing between two dedicated nodes (referred to as terminals) is used to estimate the connection reliability. Calculating this probability constitutes the 2-Terminal Reliability Problem (2-TRP), which is known to be #P-complete [20]. This complexity has motivated numerous researchers to develop approximate and exact algorithms for its solution.

One approach for designing reliable networks was proposed in [4], where the authors introduced the Resilient Path Selection Problem (RPSP). This problem considers a weighted network with two designated vertices, with costs and reliability values assigned to each arc. The objective is to identify a subnet that meets a specified reliability threshold. The authors presented an algorithm solving a nonlinear relaxation method. However, the reliability of subnets was computed imprecisely, potentially leading to infeasible solutions.

In a recent publication [16], the authors introduced the Efficient 2-Terminal Reliability Problem (E2TRP), an adaptation combining elements of RPSP and 2-TRP for reliable supply chain design. Their developed Branch-and-Price algorithm addresses the reliability calculation and feasibility challenges identified in [4]. Computational experiments demonstrated the method's ability to identify both optimal and near-optimal solutions. In this work, we extend that approach by introducing new branching strategies, advanced pricing problem solution techniques, and the development of novel valid inequalities and primal heuristics.

2. Related work

The 2-Terminal Reliability Problem (2-TRP) is a fundamental approach for assessing network reliability. Its main objective is to compute the probability that a route exists between two designated nodes in a network, with arcs assigned reliability (survival probability) values. As demonstrated in [20], 2-TRP is a #P-complete problem, which has spurred the development of numerous approximate and exact solution methods.

The first algorithmic results for 2-TRP date back to the classic paper by E. F. Moore and C. E. Shannon [12], who proposed the well-known recursive exact formula for this probability. Furthermore, [17] proposed an exact linear-time algorithm for series-parallel graphs. For large-scale graphs, the approach in [21] employed Monte Carlo methods to estimate the probability of route existence. More recently, [6] introduced a fully polynomial-time randomized approximation scheme (FPRAS) tailored for directed acyclic graphs (DAGs). However, since 2-TRP focuses only on network reliability, omitting cost considerations, its direct application to supply chain analysis remains limited.

Another strategy to enhance supply chain reliability involves constructing backup routes. Disjunct and low-cost alternative paths contribute to increased resiliency while maintaining low operating costs. For instance, [13] ensured the resiliency of primary routes by incorporating independent backup routes. For more complex supply chain topologies, approaches based on independent production plans were proposed in [14, 15]. Nonetheless, these methods face limitations: as shown in [4], solutions relying only on backup routes do not always guarantee optimality or satisfy reliability constraints fully.

It is also worth noting the works [7, 8] where the authors consider Network Design Problem with Vulnerability Constraints (NDPVC). The aim of NDPVC is to designs reliable telecommunication network with respect to length bounds of communication paths between units of commodity pairs and maintains network connectivity after the failure of any k links. The authors proposed integer linear programming formulation and several branch-and-cut algorithms. Later, in another work [2], the authors considered extension of NDPVC that includes edges with assigned reliability. The goal of this extension is to design a network that is robust to possible simultaneous edge failures, which have a certain probability of occurrence.

Additionally, [18] introduced an approach for the reliability network design problem that aims to identify minimal-cost subgraphs connecting two specified nodes using Monte Carlo-generated scenario sets. A key limitation of this method is the restricted number of scenarios, which constrains the accuracy of reliability assessments. To overcome this, [4] proposed the Resilient Path Selection Problem, approximated with a Branch-and-Price algorithm. While this approach avoids scenario-based methods, the nonlinearity of the problem and approximate reliability calculations often led to infeasible solutions. These issues were addressed in [16], where the authors applied Dantzig–Wolfe decomposition combined with column generation. However, the algorithm's efficiency deteriorated under conditions of higher graph density and stricter reliability constraints.

By extending the approach of [16] and recent advances in routing problem analysis [10, 11], we propose several enhancements to the original algorithm:

- novel heuristics and branching strategies to accelerate convergence;
- a new approach for solving the nonlinear integer pricing problem;
- valid constraints designed to reduce solution gaps.

This paper is organized as follows. In Section 3, we place the E2TRP mathematical statement. Section 4 presents the mathematical programming models underpinning our enhanced Branch-and-price (BnP) algorithm, including the original E2TRP formulation, probability relaxation method, and application of the Dantzig–Wolfe decomposition technique. Section 5 introduces our extensions to the Branch-and-price algorithm, such as more efficient subproblem employed in column

generation, hybrid branching techniques, primal heuristics, and valid constraints based on s-t graph cuts. Numerical results demonstrating the effectiveness of our approach are provided in Section 6. Finally, Section 7 summarizes our findings and discusses the future directions.

3. Problem statement

An instance of E2TRP is specified by a tuple $(G, s, t, \mathfrak{F})$, where

- $G = (V, E, c, p)$ is an arc-weighted digraph such that
  - the node set V contains designated supplier s and consumer t nodes;
  - for any arc $e \in E$, the weighting functions $c : E \to \mathbb{R}$ and $p : E \to [0, 1]$ assign cost c_e and reliability p_e, respectively;
- the value $0 < \mathfrak{F} < 1$ is an unreliability threshold.

Our approach is based on the generalized Erdös–Rényi framework, where each arc $e \in E$ fails independently with probability $q_e = 1 - p_e$.

A feasible solution of E2TRP is a subgraph $G' = (V', E') \subset G$, $\{s, t\} \subset V'$ containing an s-t path with probability at least $1 - \mathfrak{F}$.

The goal is to find a feasible subgraph of minimal $\text{cost}(G')$ where

$$\text{cost}(G') = \sum_{e \in E'} c_e.$$

To compute the graph G unreliability value, we employ the well-known Moore–Shannon formula:

$$(G) = q_e \mathbb{P}(G_1) + p_e \mathbb{P}(G_2), \tag{3.1}$$

where $G_1 = G \setminus \{e\}$, G_2 is a subgraph where arc e is totally reliable.

4. Mixed-integer models

The original formulation of the E2TRP was proposed in [16]. The authors relied on Dantzig–Wolfe technique to decompose the original problem to the corresponding Master Problem (MP) and subproblem. In this paper, we enhance our resent results by proposing the novel approach for solving pricing problem. For the sake of convenience, we place the description of our algorithm below with respect to the terminology proposed in [5].

4.1. Original formulation

The original integer program contains the following decision variables:

- ξ_e is a binary variable where $\xi_e = 1$ if arc $e \in E'$;
- x_e^π is a binary variable where $x_e^\pi = 1$ if arc e belongs to route π;
- x_0^π is a binary variable where $x_0^\pi = 1$ if route π belongs to a solution G'.

In addition, we employ the following technical notation: $\mathbf{x}^\pi = [x_0^\pi, x_e^\pi : e \in E]$. The integer model has the following form:

$$(\text{IP}): \ \min \sum_{e\in E} c_e \xi_e, \tag{4.1}$$

$$s.t. \quad \mathbb{P}(G') \le \mathfrak{F}, \tag{4.2}$$

$$x_e^\pi \le \xi_e \quad \begin{pmatrix} e \in E \\ \pi \in \mathfrak{R} \end{pmatrix}, \tag{4.3}$$

$$x_e^\pi \le x_0^\pi \quad (e \in E), \tag{4.4}$$

$$\sum_{(i,j)\in E} x_{(i,j)}^\pi - \sum_{(j,i)\in E} x_{(j,i)}^\pi = \begin{cases} x_0^\pi, & i = s, \\ -x_0^\pi, & i = t, \\ 0, & \text{otherwise}, \end{cases} \qquad (\pi \in \mathfrak{R}), \tag{4.5}$$

$$\mathbf{x}^\pi \ne \mathbf{x}^\sigma \quad (\pi, \sigma \in \mathfrak{R}) \tag{4.6}$$

$$x_e^\pi,\ x_0^\pi,\ \xi_e \in \{0,1\} \quad \begin{pmatrix} e \in E \\ \pi \in \mathfrak{R} \end{pmatrix}.$$

In (4.1), we minimize the cost of the constructed subgraph G'. Equation (4.2) defines the unreliability constraint. Equations (4.3) and (4.4) establish relationship between decision variables. Equation (4.5) is a flow conservation constraint of any route $\pi \in \mathfrak{R}$. Finally, equation (4.6) guarantees uniqueness of decision variables $\mathbf{x}^\pi$.

As it can be seen, the model IP appears to be non-linear because of equation (4.2). To address this issue, the authors of [4] and [16] apply the so-called probability relaxation.

Theorem 1. *Let $\mathbb{P}(\pi_i)$ be the failure probability of route π_i and $\mathbb{P}(\pi_1, \pi_2, \cdots, \pi_k)$ denotes the joint failure probability of family $\{\pi_1, \pi_2, \cdots, \pi_k\}$. Then,*

$$\prod_{i=1}^{k} \mathbb{P}(\pi_i) \le \mathbb{P}(\pi_1, \pi_2, \cdots, \pi_k).$$

P r o o f. Let $\mathfrak{R} = \{\pi_1, \pi_2, \ldots, \pi_m\}$ be a set of routes consisting of n arcs and $\mathbb{P}(\mathfrak{R})$ its join failure probability. We proof the statement by induction on the number of arcs n. The number of routes is assumed to be arbitrary.

Base case. Let $n = 1$ and all π_i, $i = \overline{1,m}$ consist of exactly one arc e whose probability of failure is $p_e \in (0,1)$. Then, $\mathbb{P}(\pi_i) = p_e$, $\mathbb{P}(\mathfrak{R}) = p_e$, and, therefore,

$$\mathbb{P}(\mathfrak{R}) = p_e \ge \prod_{i=1}^{m} \mathbb{P}(\pi_i) = p_e^m.$$

Induction step. Let the statement be valid for an arbitrary number of routes and for n arcs. Prove the statement for the case of $n+1$ arcs. Let $\mathfrak{R}$ be a set of routes including $n+1$ arcs in total. For an arbitrary arc e, we partition set $\mathfrak{R} = \mathfrak{R}_e \cup \bar{\mathfrak{R}}_e$ where $\mathfrak{R}_e$ is a subset of routes containing e, $\mathfrak{R}_e^o \subset \mathfrak{R}_e$, $\mathfrak{R}_e^o = \{r \in \mathfrak{R}_e,\ r \ne e\}$, and $\bar{\mathfrak{R}}_e$ is a subset of routes that do not contain e.

By the formula of total probability we have

$$\mathbb{P}(\mathfrak{R}) = p_e \cdot \mathbb{P}(\bar{\mathfrak{R}}_e) + (1-p_e) \cdot \mathbb{P}(\hat{\mathfrak{R}}) \ge p_e \prod_{\pi_i \in \bar{\mathfrak{R}}_e} \mathbb{P}(\pi_i) + (1-p_e) \prod_{\pi_i \in \mathfrak{R}_e^o} \mathbb{P}(\hat{\pi}_i)$$

$$\ge p_e \prod_{\pi_i \in \bar{\mathfrak{R}}_e} \mathbb{P}(\pi_i) + (1-p_e) \prod_{\pi_i \in \bar{\mathfrak{R}}_e} \mathbb{P}(\pi_i) \cdot \prod_{\pi_i \in \mathfrak{R}_e^o} \mathbb{P}(\hat{\pi}_i) = \prod_{\pi_i \in \bar{\mathfrak{R}}_e} \mathbb{P}(\pi_i) \left(p_e + (1-p_e) \prod_{\pi_i \in \mathfrak{R}_e^o} \mathbb{P}(\hat{\pi}_i) \right),$$

where $\hat{\mathfrak{R}}$ is a family of non-empty routes $\hat{\pi}_i = \pi_i \setminus \{e\}$ for $\pi_i \in \mathfrak{R}_e^o$.

Here, we bound

$$\mathbb{P}(\bar{\mathfrak{R}}_e) \geq \prod_{\pi_i \in \bar{\mathfrak{R}}_e} \mathbb{P}(\pi_i), \quad \mathbb{P}(\hat{\mathfrak{R}}) \geq \prod_{\pi_i \in \mathfrak{R}_e^o} \mathbb{P}(\hat{\pi}_i)$$

by the induction hypothesis.

Prove separately that

$$\left(p_e + (1-p_e) \prod_{\pi_i \in \mathfrak{R}_e^o} \mathbb{P}(\hat{\pi}_i)\right) \geq \prod_{\pi_i \in \mathfrak{R}_e^o} \mathbb{P}(\pi_i).$$

For that, consider an expression

$$\big(p_e + (1-p_e)\cdot(1-x_1)\cdot\ldots\cdot(1-x_k)\big) - \big(1-(1-p_e)\cdot x_1\big)\cdot\ldots\cdot\big(1-(1-p_e)\cdot x_k\big),$$

where

$$x_i = \prod_{b \in \pi_i \setminus \{e\}} (1-p_b), \quad k = |\mathfrak{R}_e^o|.$$

Denote as $c = 1-p_e$, $0 < c < 1$. Then, we have

$$f(x_1, x_2, \ldots, x_k) = 1 - c + c(1-x_1)\cdot\ldots\cdot(1-x_k) - (1-cx_1)(1-cx_2)\cdot\ldots\cdot(1-cx_k).$$

The function $f\colon (0,1)^k \to \mathbb{R}_+$. It is easy to see, that

$$f(0,0,\ldots,0) = 1 - c + c - 1 = 0$$

and

$$\frac{\partial f}{\partial x_i} = -c \prod_{j=1, j\neq i}^{k} (1-x_j) + c \prod_{j=1, j\neq i}^{k} (1-cx_j) = c\left[\prod_{j=1, j\neq i}^{k} (1-cx_j) - \prod_{j=1, j\neq i}^{k} (1-x_j)\right] \geq 0$$

for all $x \geq 0$ and strictly great zero when at least one $x_j > 0$. Hence, $\nabla f(x) > 0$ for any $x \in (0,1)^k$, $x \neq 0$, and by Lagrange Theorem we have $f(x_1, \ldots, x_k) > 0$ for any $x \in (0,1)^k$, $x \neq 0$.

Continuing the proof, we have

$$\begin{aligned}\prod_{\pi_i \in \bar{\mathfrak{R}}_e} \mathbb{P}(\pi_i) \left(p_e + (1-p_e) \prod_{\pi_i \in \mathfrak{R}_e^o} \mathbb{P}(\hat{\pi}_i)\right) &\geq \prod_{\pi_i \in \bar{\mathfrak{R}}_e} \mathbb{P}(\pi_i) \cdot \prod_{\pi_j \in \mathfrak{R}_e^o} \mathbb{P}(\pi_j) \\ &\geq \prod_{\pi_i \in \bar{\mathfrak{R}}_e} \mathbb{P}(\pi_i) \cdot \prod_{\pi_j \in \mathfrak{R}_e} \mathbb{P}(\pi_j) = \prod_{\pi_i \in \mathfrak{R}} \mathbb{P}(\pi_i).\end{aligned}$$

The proof is follows. □

Let $\pi_1, \pi_2, \cdots, \pi_k$ be all possible routes of graph G'. Then,

$$\prod_{i=1}^{k} \mathbb{P}(\pi_i) \leq \mathbb{P}(G').$$

Therefore, we proceed with probability relaxation by replacing equation (4.2) with the following:

$$\sum_{\pi \in \mathfrak{R}} \tilde{\mathbb{P}}(\pi) x_0^{\pi} \leq \tilde{\mathfrak{F}}, \quad \tilde{\mathbb{P}}(\pi) = \log(\mathbb{P}(\pi)), \quad \tilde{\mathfrak{F}} = \log(\mathfrak{F}).$$

4.2. Dantzig–Wolfe decomposition

To address the exponential number of constraints, we apply the classic Dantzig–Wolfe decomposition followed by the column generation technique [5, 19].

According to this approach, for any route $\pi \in \mathfrak{R}$, $\mathbf{x}^\pi \in \mathcal{D}$, where

$$\mathcal{D} = \left\{ \mathbf{x} \in \{0,1\}^{|E|+1} \;\middle|\; \begin{array}{c} \sum\limits_{(i,j)\in E} x_{(i,j)} - \sum\limits_{(j,i)\in E} x_{(j,i)} = \begin{cases} x_0, & i = s, \\ -x_0, & i = t, \\ 0, & \text{otherwise}, \end{cases} \\ x_e \le x_0 \quad (e \in E) \end{array} \right\}.$$

By the Minkowski–Weyl theorem, we can represent any vector $\mathbf{x} \in \operatorname{conv}(\mathcal{D})$ as a convex combination of the extreme points $p \in P$ of this polytope:

$$\mathbf{x} = \sum_{p\in P} \mathbf{x}^p \lambda_p, \quad \sum_{p\in P} \lambda_p = 1, \quad \lambda_p \ge 0.$$

In our case, the set of extreme points P, the set $\mathcal{D}$ and the set of all s-t paths $\mathfrak{R}$ are the same. Therefore, $\mathbf{x}^p = \mathbf{x}^\pi$, where $p \in P$ and $\mathbf{x}^\pi \in \mathcal{D}$. In addition, variables λ_p and $\mathbf{x}_0^p$ denote path using in solution, i.e. $\lambda_p = \mathbf{x}_0^p$ for some $p \in P$. We obtain the following formulation after substitution of $\mathbf{x}^\pi$ into the original model with the relaxed probability constraint:

$$\begin{aligned} &\min \sum_{e\in E} c_e \xi_e, \\ s.t. \quad &\sum_{p\in P} \tilde{\mathbb{P}}(\mathbf{x}^p)\lambda_p \le \tilde{\mathfrak{F}}, \\ &x_e^p \lambda_p \le \xi_e \quad \begin{pmatrix} e \in E \\ p \in P \end{pmatrix}, \\ &\lambda_p \in \{0,1\} \quad (p \in P), \\ &\xi_e \in \{0,1\} \quad (e \in E). \end{aligned} \tag{4.7}$$

To reduce the number of constraints, we aggregate equations (4.7). Additionally, the following valid inequality is added in order to speedup convergence:

$$\sum_{p\in P} \lambda_p \ge k_{\min}, \quad k_{\min} = \left\lceil \frac{\tilde{\mathfrak{F}}}{\tilde{\mathbb{P}}(p^*)} \right\rceil,$$

where p^* denotes the most reliable s-t path in graph G.

As the final result, we obtain the following Integer Master Problem:

$$\begin{aligned} &(\text{IMP}): \ \min \sum_{e\in E} c_e \xi_e, \\ s.t. \quad &\sum_{p\in P} \tilde{\mathbb{P}}(\mathbf{x}^p)\lambda_p \le \tilde{\mathfrak{F}}, && (4.8) \\ &\sum_{p\in P} x_e^p \lambda_p \le M\xi_e \quad (e \in E), && (4.9) \\ &\sum_{p\in P} \lambda_p \ge k_{\min}, && (4.10) \\ &\lambda_p \in \{0,1\} \quad (p \in P), && (4.11) \\ &\xi_e \in \{0,1\} \quad (e \in E), && (4.12) \end{aligned}$$

where M is a sufficiently large constant, which sets upper bound on the use of arcs. Therefor, model may be unfeasible for small M values, and computational difficulties may occur for large M. For our experiment benchmark, we use $M = 1000$, which was obtained empirically. This value is sufficient to guarantee solution feasibility and obtain sufficient approximation after integrality relaxation.

5. Branch-and-price algorithm

Our branch-and-price algorithm is based on the methodology proposed in [16]. At each node of the branch-and-bound tree, we solve the linear relaxation of the Integer Master Problem (IMP) using column generation. Since IMP represents probability relaxation of the initial problem, each its integer solution is called feasible (incumbent) if it satisfies the unreliability constraint 4.2. Otherwise, a branching procedure performed.

5.1. Column generation

To solve IMP with an exponential number of variables $\lambda_p, (p \in P)$, we employ the column generation technique. Specifically, we relax the integrality constraints (4.11) and (4.12) and solve the Restricted Master Problem (RMP) over a subset of paths $P' \subset P$. To ensure feasibility of the RMP, artificial variables are introduced. New columns are then generated iteratively by solving a pricing problem to extend the subset P'.

Let α, β_e and γ denote the dual variables associated with constraints (4.8), (4.9) and (4.10), respectively. Then, the Dual problem of RMP (DRMP) is formulated as follows:

$$\begin{gathered}
\mathrm{DRMP}(P') : \max -\tilde{\mathfrak{F}}\alpha + k_{\min}\gamma \\
s.t. \\
-\alpha\tilde{\mathbb{P}}(\mathbf{x}^p) - \sum_{e\in E}\beta_e x_e^p + \gamma \le 0 \quad (p \in P'), \\
M\beta_e - c_e \le 0 \quad (e \in E), \quad \alpha \ge 0, \quad \beta_e \ge 0 \quad (e \in E), \quad \gamma \ge 0.
\end{gathered}$$

Given an optimal solution $(\bar{\alpha}, \bar{\beta}_e, \bar{\gamma})$ of the dual restricted master problem $\mathrm{DRMP}(P')$, we check whether there exists a path $p \in P \setminus P'$ with negative reduced cost, i.e., if the following inequality holds:

$$\bar{\alpha}\tilde{\mathbb{P}}(\mathbf{x}^p) + \sum_{e\in E}\bar{\beta}_e x_e^p - \bar{\gamma} < 0.$$

If such a path exists, it is added to the RMP, and the column generation process continues. If no paths are found, the column generation procedure stops.

The path searching process is complicated because of the nonlinearity of the term $\bar{\alpha}\tilde{\mathbb{P}}(\mathbf{x}^p)$. To address this issues, we consider two cases: when $\bar{\alpha} = 0$ and when $\bar{\alpha} > 0$.

In the case where $\bar{\alpha} = 0$, we formulate and solve the following auxiliary problem (AUX):

$$\begin{gathered}
(\mathrm{AUX}) : \min \sum_{e\in E} x_e\bar{\beta}_e, \\
s.t. \\
\sum_{(i,j)\in E} x_{(i,j)} - \sum_{(j,i)\in E} x_{(j,i)} = \begin{cases} 1, & i = s, \\ -1, & i = t, \\ 0, & \text{otherwise}, \end{cases}
\end{gathered} \tag{5.1}$$

$$\sum_{e \in E(p)} x_e \leq |E(p)| - 1 \quad (p \in P'), x_e \in \{0,1\} \quad (e \in E).$$

Here, x_e is a binary variable indicating whether edge e is used in the solution path, $E(p)$ denotes the set of edges composing path $p \in P'$. Constraint (5.1) represents the flow conservation constraints, while equation (5.1) ensures that the newly generated path has not been previously appended to the RMP in previous iterations.

When $\bar{\alpha} > 0$, we proceed as follows:

(i) Find the path p^* minimizing the value

$$\sum_{e \in E} \bar{\beta}_e x_e^{p^*}$$

by solving the auxiliary problem (AUX). If the reduced cost of p^* is negative, then process terminates. Otherwise, we proceed to step (ii).

(ii) Since

$$\sum_{e \in E} \bar{\beta}_e x_e^{p^*}$$

is minimal, it follows that for any other path

$$q \in P \setminus P', \quad \sum_{e \in E} \bar{\beta}_e x_e^{q} \geq \sum_{e \in E} \bar{\beta}_e x_e^{p^*}.$$

Hence, to find a path q with negative reduced cost, the following condition on the nonlinear term must hold:

$$\bar{\alpha}\tilde{\mathbb{P}}(x^q) \leq \bar{\alpha}\tilde{\mathbb{P}}(x^{p^*}).$$

To implement this, we add the constraint

$$\sum_{e \in E} \log(p_e) x_e^q \geq \sum_{e \in E(p^*)} \log(p_e).$$

Additionally, to prevent regenerating the same path p^*, we append the following constraint to AUX:

$$\sum_{e \in E(p^*)} x_e^q \leq |E(p^*)| - 1.$$

Afterward, we return to step (i).

The steps (i) and (ii) are repeated iteratively until either a path with negative reduced cost is found or the auxiliary problem becomes infeasible; in the latter case, column generation stops.

In practice, after identifying a path with negative reduced cost, it is often beneficial to continue the search and add multiple such columns to the RMP to accelerate the convergence of the algorithm. In this paper, we follow the same technique.

5.2. Branching scheme

As it follows from the Section 4, we employ the two-stage relaxation method to address both the integrality condition and the nonlinearity of the problem, similarly to the approach used by the authors of [16]. In the first stage, the probability condition is relaxed, and in the second stage, the integrality condition is relaxed. Accordingly, we apply two distinct branching schemes to obtain a valid integer solution.

In [16], the authors employed *most fractional branching* and *probability branching*. Most fractional branching is the simplest strategy; however, experiments [1] indicate that its performance is comparable to random selection. To enhance efficiency, we retain probability branching but replace most fractional branching with a *hybrid branching strategy* that combines *strong branching* and *pseudocost branching*.

Let $(\bar{\xi}_e, \bar{\lambda}_p)$ denote an optimal solution to the LP relaxation of the problem. If $\bar{\xi}$ is integral, we apply probability branching; otherwise, we use hybrid branching.

Probability branching. At first, we verify the satisfaction of the unreliability constraint by formula (3.1). If $\mathbb{P}(G')$ violates the unreliability constraint (4.2), we spawn two child nodes as follows:

(1) add the constraint
$$\sum_{e \in E'} \xi_e \leq |E'| - 1$$
to eliminate subgraph G' from the solution;
(2) keep subgraph G' and augment it by additional edges setting
$$\xi_e = 1 \quad (e \in E'), \quad \sum_{e \in E} \xi_e \geq |E'| + 1.$$
Additionly, we increment the $k_{\min}$ value by one.

Strong branching. This procedure estimates the potential impact of branching on the RMP objective value. For each edge $e \in E$ with fractional value $\bar{\xi}_e > 0$, two auxiliary RMPs are solved with the additional constraints $\xi_e = 0$ and $\xi_e = 1$, respectively. Let:

(1) $\bar{z}$ be the objective value of the current fractional solution $(\bar{\xi}_e, \bar{\lambda}_p)$;
(2) z_e^- and z_e^+ be the objective values of the respective auxiliary RMPs.

We compute the impact estimates as:
$$\delta_e^- = z_e^- - \bar{z}, \quad \delta_e^+ = z_e^+ - \bar{z}$$
The edge selected for branching is the one maximizing the product $\delta_e^- \times \delta_e^+$. To save computational effort, no additional column generation is performed during these auxiliary solves.

Pseudocost branching. This method exploits branching history to estimate the expected objective variations. Define:

(1) z_{parent} as the objective value at the parent node in the searching tree;
(2) ξ_e^{parent} as the value of ξ_e at the parent node;
(3) z_{child}^- and z_{child}^+ as the objective values of the child nodes after branching on ξ_e.

The *pseudocosts* for edge e are defined as:
$$\zeta^-(e) = \frac{z_{child}^- - z_{parent}}{\xi_e^{parent}}, \quad \zeta^+(e) = \frac{z_{child}^+ - z_{parent}}{1 - \xi_e^{parent}},$$
which represent the objective change per unit change in ξ_e for fixing $\xi_e = 0$ and $\xi_e = 1$, respectively. Let $\zeta_i^-(e),\ \zeta_i^+(e),\ i \in \{1, 2, \dots n\}$ be the pseudocost values collected during the branching process. The average pseudocosts are
$$\psi^-(e) = \frac{\zeta_1^-(e) + \dots + \zeta_n^-(e)}{n}, \quad \psi^+(e) = \frac{\zeta_1^+(e) + \dots + \zeta_n^+(e)}{n}.$$

The estimates of objective changes are then computed as

$$\delta_e^- = \psi^-(e)\bar{\xi}_e, \quad \delta_e^+ = \psi^+(e)(1-\bar{\xi}_e).$$

Finaly, the edge maximizing $\delta_e^- \times \delta_e^+$ is selected.

Hybrid branching. At the beginning of the Branch-and-Price algorithm, the branching history is limited or nonexistent, which precludes the immediate use of pseudocost branching. Therefore, edges are branched according with the strong branching strategy until sufficient pseudocost data has been collected (i.e., the edge has been branched on a predefined minimum number of times). Afterwards, pseudocost branching is applied for those edges.

5.3. Primal heuristics

The authors of [16] employ *start greedy* and *local search* primal heuristics. In our algorithm, we adopt these heuristics and enhance them with two additional approaches:

(1) *Start Greedy.* This heuristic aims to find a feasible solution consisting of independent routes. It iteratively constructs the solution by searching for the shortest independent paths.

(2) *Local Search.* For each branching node, we formulate the following integer program over the set of routes P':

$$\min \sum_{e\in E} c_e\xi_e,$$

$$\sum_{p\in P'} \tilde{\mathbb{P}}(\mathbf{x}^p)\lambda_p \le \tilde{\mathfrak{F}},$$

$$\sum_{p\in P'} x_e^p\lambda_p \le \xi_e \quad (e\in E), \tag{5.2}$$

$$\xi_e \in \{0,1\}, \quad \lambda_p \in \{0,1\}. \tag{5.3}$$

Due to constraints (5.2) and (5.3), the solution contains only independent paths,ensuring feasibility for the E2TRP problem. Whenever this heuristic successfully finds an incumbent solution, we update the upper bound.

(3) *Shortest Path Check.* Before the Branch-and-Price process, this heuristic verifies whether the shortest s-t path is a feasible solution. If so, the shortest path corresponds to an exact solution.

(4) *Complementary Column Generation* [5]. This heuristic is designed to accelerate convergence by generating additional columns. The objective is to find columns orthogonal to those identified by the exact method. However, in our specific application, this procedure is computationally intensive and does not guarantee finding new unique columns with negative reduced cost. Therefore, we remove the most frequently used arcs and solve the pricing problem on the reduced graph to generate new paths.

5.4. Valid inequalities for s-t-cuts in the graph

To speedup the convergency of the algorithm, we employ valid inequalities based on s-t cuts. Let $G' = (V', E')$ be a solution subgraph, $C \subset E$ be an s-t cut of the graph G, and define

$$C' = \{e \in C |\ \xi_e = 1\}$$

as the subset of arcs in the cut that are selected in the solution. The unreliability of G' is given by

$$\mathbb{P}(G') = \prod_{e\in C'} q_e + \Big(1 - \prod_{e\in C'} q_e\Big)\mathbb{P}(G' |\text{ at least one arc in } C' \text{ survives}),$$

where q_e denote the failure probability of arc e. Since $\mathbb{P}(G') \leq \mathfrak{F}$, it follows that:

$$\prod_{e \in C'} q_e \leq \mathfrak{F}.$$

Taking the logarithm of both sides yields the valid inequality

$$\sum_{e \in C'} \log(q_e) \leq \tilde{\mathfrak{F}}.$$

Taking in account the decision variables ξ_e, it becomes

$$\sum_{e \in C} \log(q_e)\xi_e \leq \tilde{\mathfrak{F}}. \tag{5.4}$$

Constraint (5.4) involves only the ξ_e variables and does not affect the pricing problem. In our algorithm, we add these constraints for the arcs outgoing from node s, incoming to node t, and those forming a minimal s-t cut in graph G. We recompute and append these constraints before the column generation process at every node of the search tree to maintain their validity during branching and thereby improve convergence.

6. Numerical evaluation

6.1. Instance benchmark

We evaluate our approach using E2TRP instances from [16]. Each instance is represented as a tuple $(G, \mathfrak{F})$, where G is a graph with nodes uniformly distributed within the square $(-100, 100) \times (-100, 100)$ in the Euclidean plane, and $\mathfrak{F}$ is an unreliability threshold. Each graph G is characterized by the number of nodes $|V| \in \{20, 30\}$ and a density parameter $\gamma \in \{0.25, 0.5, 0.75\}$. The arc weights $w_e, \quad (e \in E)$ are defined as Euclidean distances, and the failure probabilities q_e are assigned from $\{5 \cdot 10^{-3}, 10^{-3}, 10^{-4}, 10^{-5}\}$. The unreliability threshold $\mathfrak{F}$ takes values from the set $\{1 \cdot 10^{-1}, 5 \cdot 10^{-2}, 1 \cdot 10^{-2}, 5 \cdot 10^{-3}, 1 \cdot 10^{-3}, 5 \cdot 10^{-4}, 1 \cdot 10^{-4}, 5 \cdot 10^{-5}, 1 \cdot 10^{-5}, 5 \cdot 10^{-6}, 1 \cdot 10^{-6}\}$. For each combination, $(|V|, \gamma, \mathfrak{F})$ there are 10 instances, resulting in a total of 660 instances in the benchmark.

6.2. Experimental setup

The results reported in [16] demonstrate that their algorithm can find exact and feasible solutions. However, the average gap and the number of instances solved to optimality deteriorate for instances with higher density γ and tighter probability thresholds $\mathfrak{F}$. Thus, instances with $|V| = 30$, $\gamma = 0.75$, and $\mathfrak{F} \leq 1 \cdot 10^{-3}$ were not solved to optimality within a 10-hour time limit.

To address this issue, we set a fixed time limit of 10 hours and compare the results of [16] (algorithm A_1) with those of our proposed algorithm A_2. Specifically, we separate experiment into two stages. At the first stage we compare A_1 and A_2 as exact algorithms, and at the second stage, as approximate algorithms. Algorithm A_2 is implemented in Python 3.9.7 using NetworkX library. For solving LP relaxations and subproblem integer programs, we employed the Gurobi MIP solver [9]. All results of the numerical evaluation are presented in Table 1.

The experiments were carried out on the 'Uran' supercomputer center of N.N. Krasovsky Institute of Mathematics and Mechanics, Ural Branch of Russian Academy of Science http://parallel.uran.ru, on Intel(R) Xeon(R) 16 core CPU E5-2697 v4 @2.30 GHz 256GB RAM.

Table 1. Comparison results of algorithms A_1 and A_2. Parameters include γ (graph density), $\mathfrak{F}$ (unreliability threshold), and $|V|$ (number of nodes in the instance graph). *OPT*: Number of instances solved to optimality. *ABN*: Average number of branching nodes. avg gap, %: Average gap. The first two columns specify the instance parameters γ and $\mathfrak{F}$. The next six columns present results for instances with $|V| = 20$ nodes for both algorithms A_1 and A_2. The last six columns provide the same results for instances with $|V| = 30$. The best gap values are highlighted in bold.

		$\|V\| = 20$						$\|V\| = 30$					
		A_1			A_2			A_1			A_2		
γ	$\mathfrak{F}$	OPT	ABN	avg gap, %	OPT	ABN	avg gap, %	OPT	ABN	avg gap, %	OPT	ABN	avg gap, %
25.0	10^{-1}	10.0	3607.8	0.00	10.0	0.0	0.00	8.0	56920.6	3.14	10.0	0.0	**0.00**
	$5 \cdot 10^{-2}$	10.0	3607.8	0.00	10.0	0.0	0.00	8.0	56920.6	3.14	10.0	0.0	**0.00**
	10^{-2}	10.0	2068.6	0.00	10.0	0.0	0.00	8.0	25404.4	2.40	10.0	0.0	**0.00**
	$5 \cdot 10^{-3}$	10.0	1552.4	0.00	10.0	76.5	0.00	8.0	19153.5	1.17	10.0	137.0	**0.00**
	10^{-3}	8.0	12588.3	1.26	10.0	2236.3	**0.00**	8.0	26404.5	2.06	10.0	375.6	**0.00**
	$5 \cdot 10^{-4}$	10.0	18059.1	0.00	10.0	1281.6	0.00	8.0	25947.3	1.70	10.0	264.3	**0.00**
	10^{-4}	7.0	73885.0	4.85	9.0	8217.4	**2.05**	5.0	40336.6	20.28	9.0	18335.2	**3.13**
	$5 \cdot 10^{-5}$	7.0	49897.9	6.64	10.0	3104.4	**0.00**	4.0	42185.3	20.42	10.0	12490.4	**0.00**
	10^{-5}	5.0	56711.5	11.68	10.0	8992.7	**0.00**	3.0	36239.0	24.77	7.0	21707.0	**4.71**
	$5 \cdot 10^{-6}$	3.0	67474.4	14.33	8.0	9215.3	**1.65**	2.0	34515.2	29.05	7.0	14025.7	**3.96**
	10^{-6}	3.0	59927.1	18.75	8.0	13990.2	**3.10**	2.0	27162.4	31.41	7.0	12738.4	**4.63**
50.0	10^{-1}	10.0	9182.6	0.00	10.0	0.0	0.00	4.0	52933.4	8.82	10.0	0.0	**0.00**
	$5 \cdot 10^{-2}$	10.0	9182.6	0.00	10.0	0.0	0.00	4.0	52933.4	8.82	10.0	0.0	**0.00**
	10^{-2}	10.0	8444.4	0.00	10.0	0.0	0.00	4.0	36123.0	10.92	10.0	0.0	**0.00**
	$5 \cdot 10^{-3}$	10.0	6609.2	0.00	10.0	7.4	0.00	4.0	27190.2	13.12	10.0	2254.9	**0.00**
	10^{-3}	7.0	21457.3	3.98	10.0	1624.9	**0.00**	4.0	23955.8	12.56	10.0	2086.1	**0.00**
	$5 \cdot 10^{-4}$	8.0	17351.9	2.55	10.0	740.4	**0.00**	3.0	28477.6	15.08	10.0	1841.9	**0.00**
	10^{-4}	6.0	26172.1	5.68	10.0	1194.7	**0.00**	2.0	25629.1	32.10	8.0	4959.0	**11.10**
	$5 \cdot 10^{-5}$	7.0	22185.0	5.60	10.0	893.6	**0.00**	1.0	25899.9	38.36	7.0	5786.9	**12.66**
	10^{-5}	5.0	29727.4	12.81	9.0	5029.4	**1.91**	1.0	32232.9	39.06	5.0	11111.6	**14.12**
	$5 \cdot 10^{-6}$	5.0	30477.4	16.73	8.0	7679.5	**3.38**	0.0	35114.3	42.68	5.0	11478.7	**12.98**
	10^{-6}	4.0	28915.9	22.84	8.0	17374.2	**5.53**	0.0	21523.5	47.99	4.0	6554.9	**18.73**
75.0	10^{-1}	6.0	67308.0	7.16	10.0	0.0	**0.00**	1.0	57790.2	44.45	10.0	0.0	**0.00**
	$5 \cdot 10^{-2}$	6.0	67308.0	7.16	10.0	0.0	**0.00**	1.0	57810.4	44.44	10.0	0.0	**0.00**
	10^{-2}	6.0	38357.6	9.96	10.0	0.0	**0.00**	1.0	39927.0	48.41	10.0	0.0	**0.00**
	$5 \cdot 10^{-3}$	5.0	28099.6	9.15	10.0	25.6	**0.00**	1.0	30397.0	48.20	10.0	543.1	**0.00**
	10^{-3}	5.0	28796.8	10.10	10.0	1377.8	**0.00**	0.0	31808.6	50.30	10.0	1697.8	**0.00**
	$5 \cdot 10^{-4}$	5.0	27734.8	10.05	10.0	780.6	**0.00**	0.0	32276.6	49.75	10.0	3452.2	**0.00**
	10^{-4}	5.0	17916.6	21.26	7.0	1538.3	**14.58**	0.0	31572.8	50.76	9.0	2850.3	**3.50**
	$5 \cdot 10^{-5}$	2.0	24889.3	27.70	7.0	4250.0	**13.74**	0.0	31387.6	50.50	9.0	3996.7	**3.46**
	10^{-5}	2.0	22787.0	33.44	6.0	5601.4	**17.51**	0.0	24130.2	59.37	5.0	7723.5	**21.77**
	$5 \cdot 10^{-6}$	2.0	20084.4	39.91	5.0	5754.2	**19.40**	0.0	15438.0	71.39	2.0	5610.8	**36.66**
	10^{-6}	2.0	20747.1	38.87	5.0	4263.0	**18.52**	0.0	12517.4	75.03	1.0	6416.7	**38.05**

6.3. The first experiment stage

Fig. 1 and 2 illustrate the number of instances solved to optimality for graphs with 20 and 30 nodes, respectively.

For $|V| = 20$, algorithm A_1 solved to optimality 211 instances, while A_2 solved 300. For $|V| = 30$, A_1 solved 95 instances compared to 275 for A_2. Overall, A_1 solved to optimality 306 instances, whereas A_2 solved 575, demonstrating that A_2 outperforms A_1 by nearly a factor of two, with all instances solved by A_1 also being solved by A_2.

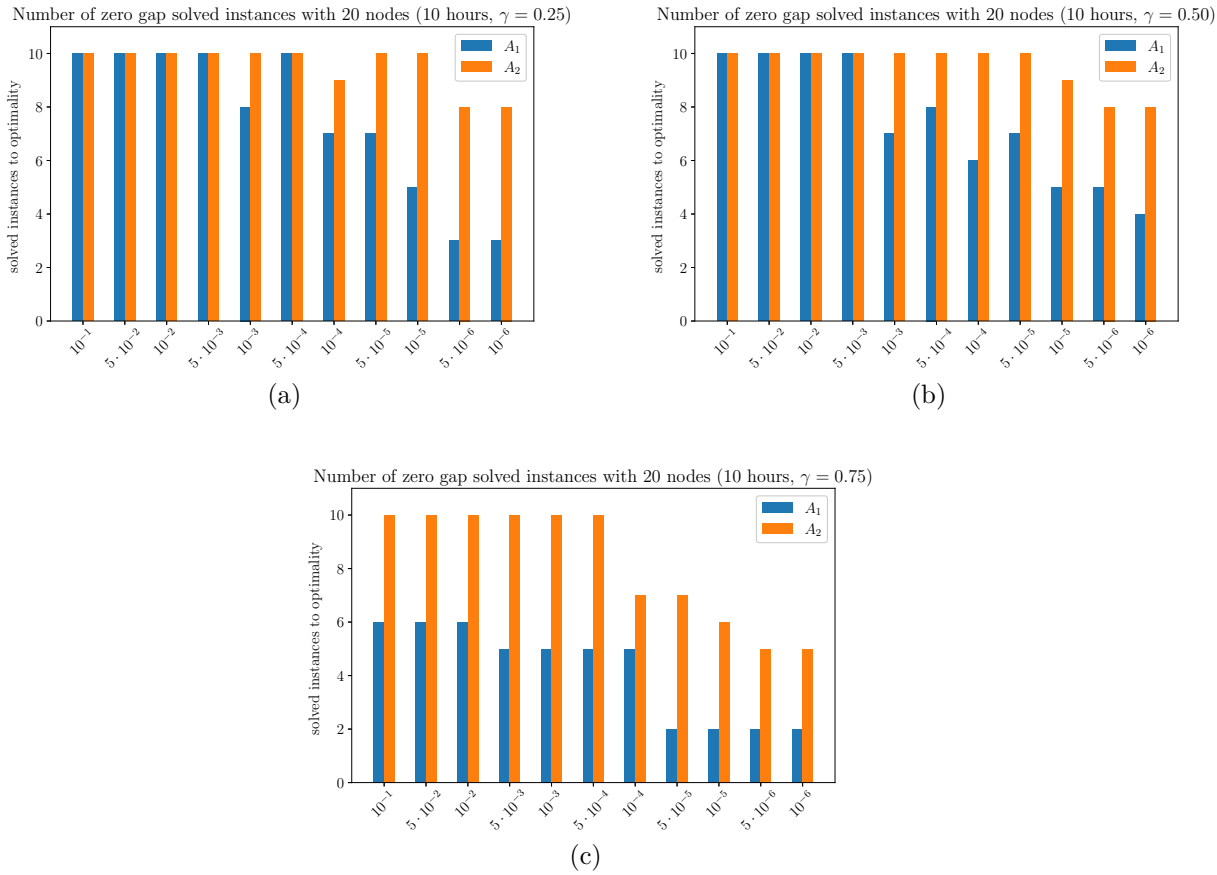

Figure 1. Number of instances solved to optimality by algorithms A_1 and A_2 for $|V| = 20$ within a 10-hour time limit, under varying threshold values. Fig. 1a depicts the results for $\gamma = 0.25$, Fig. 1b for $\gamma = 0.5$, and Fig. 1c for $\gamma = 0.75$.

6.4. The second experiment stage

To evaluate the accuracy of the algorithms, we use the average gap measure, computed as $(UB - LB)/UB$, where UB is the incumbent upper bound and LB is the corresponding lower bound. Table 1 summarizes the computational results. In most cases, for the cases where the instance was not solved to optimality by A_1, A_2 exhibits either a better average gap or manages to solve the entire group to optimality. Notably, for the most challenging instances ($|V| = 30, \gamma = 0.75$, $\mathfrak{F} = 5 \cdot 10^{-6}$) and ($|V| = 30, \gamma = 0.75, \mathfrak{F} = 10^{-6}$), the average gap achieved by A_2 is nearly half of that obtained by A_1. Additionally, the average number of branching nodes decreases significantly. For $|V| = 20$, the maximum average number of branch-and-bound nodes is 73885.0 for A_1 and 17374.2 for A_2. Similar reductions are observed for $|V| = 30$, with maximum average branch-and-bound nodes of 57810.4 for A_1 and 21707.0 for A_2.

7. Conclusion

In this paper, we extended the algorithmic analysis of the Efficient 2-Terminal Reliability Problem (E2TRP). We proposed the improved branch-and-price (BnP) algorithm, augmented with new valid inequalities based on s-t cuts in the graph G, as well as new primal heuristics, branching schemes, and an iterative method for solving the nonlinear integer pricing problem. The results of

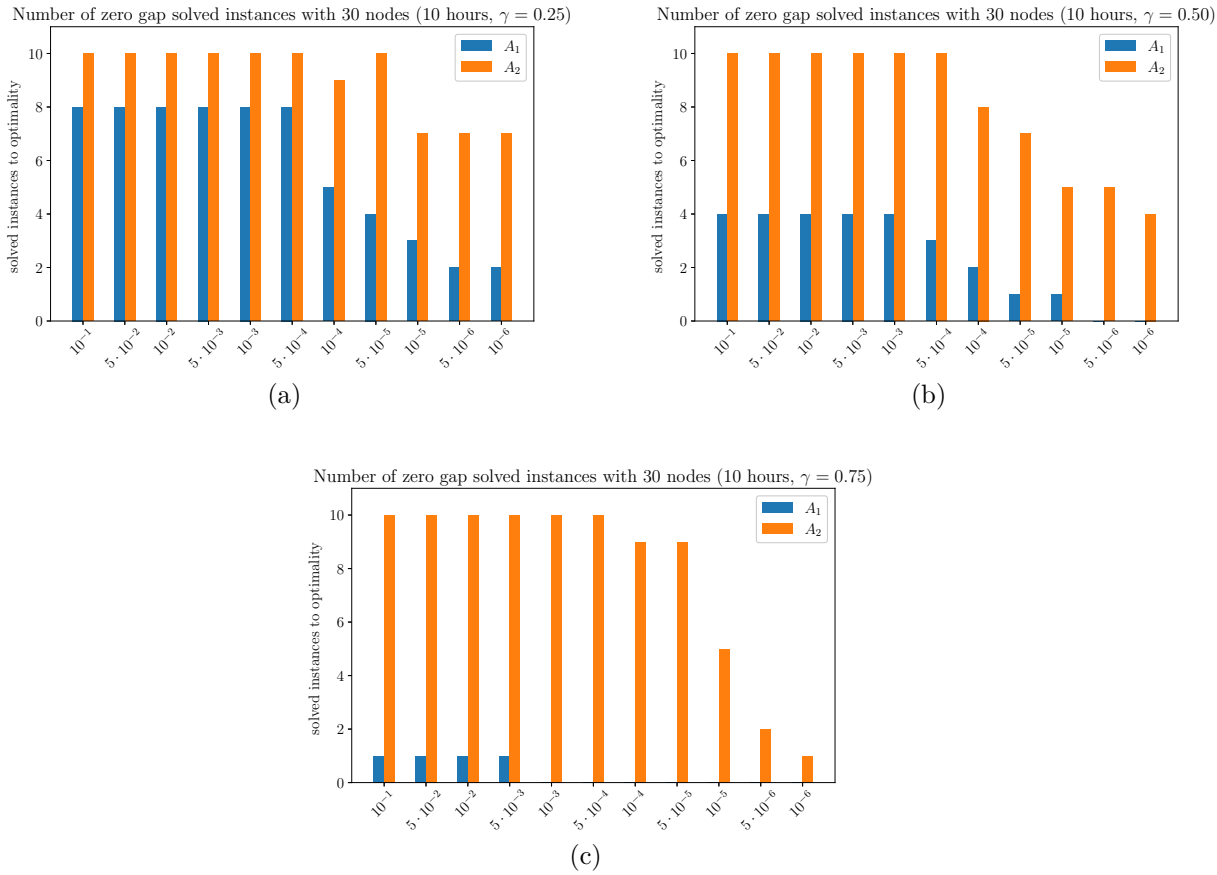

Figure 2. The number of instances solved to optimality by algorithms A_1 and A_2 for $|V| = 30$ with different threshold values in 10 hours, where 2a is number of solved instances with $\gamma = 0.25$, 2b is number of solved instances with $\gamma = 0.5$, and 2c is number of solved instances with $\gamma = 0.75$.

our comparative numerical evaluation demonstrate significant improvements in both the average number of search tree nodes and the average gap values.

One of the most promising direction of further research is to improve reliability evaluation. Theorem 1 provides a necessary condition for a feasible integer solution. For this reason, we use the Moore–Shannon formula 3.1 to accurately evaluate graph reliability. Unfortunately, this recursive formula is computationally expansive and can become the most resource-consuming part of BnP for large graph instances. Therefore, advanced 2-TRP solution techniques, such as exact algorithms for graphs with specific structures, Monte Carlo methods, approximate algorithms, or using upper and lower bounds, can improve the performance of the algorithm.

Another promising research direction is bi-criteria case of E2TRP. Constant $\mathfrak{F}$ defines upper bound for solution's unreliability and constraints solution space. Optimization over graph cost and reliability provides general solution for application, which we postpone for the future work.

REFERENCES

1. Achterberg T., Koch Th., Martin A. Branching rules revisited. *Oper. Res. Lett.*, 2005. Vol. 33, No. 1. P. 42–54. DOI: 10.1016/j.orl.2004.04.002
2. Arslan O., Laporte G. Network design with vulnerability constraints and probabilistic edge reliability. *Networks*, 2024. Vol. 84, No. 2. P. 181–199. DOI: 10.1002/net.22222

3. Bouchery Y., Corbett C. J., Fransoo J. C., Tan T. (eds.) *Sustainable Supply Chains: A Research-Based Textbook on Operations and Strategy*. Springer Ser. Supply Chain Manag. Cham: Springer, 2024. 553 p. DOI: 10.1007/978-3-031-45565-0
4. Casazza M., Ceselli A. Optimization algorithms for resilient path selection in networks. *Comput. Oper. Res.*, 2021. Vol. 128. Art. no. 105191. DOI: 10.1016/j.cor.2020.105191
5. Desrosiers J., Lübbecke M., Desaulniers G., Gauthier J. B. *Branch-and-price*. Montréal, Canada: Les Cahiers du GERAD, 2024. 657 p.
6. Feng W., Guo H. An FPRAS for two terminal reliability in directed acyclic graphs. In: *Leibniz Int. Proc. Informatics (LIPIcs), vol. 297: 51st Int. Colloquium on Automata, Languages, and Programming (ICALP 2024)*. Germany: Dagstuhl, 2024. P. 62:1–62:19. DOI: 10.4230/LIPIcs.ICALP.2024.62
7. Gouveia L., Leitner M. Design of survivable networks with vulnerability constraints. *European J. Oper. Res.*, 2017. Vol. 258, No. 1. P. 89–103. DOI: 10.1016/j.ejor.2016.09.003
8. Gouveia L., Joyce-Moniz M., Leitne M. Branch-and-cut methods for the Network Design Problem with Vulnerability Constraints. *Comput. Oper. Res.*, 2018. Vol. 91. P. 190–208. DOI: 10.1016/j.cor.2017.10.005
9. *Gurobi Optimization, LLC.* Gurobi Optimizer Reference Manual, vers. 11.0. 2024. URL: https://www.gurobi.com
10. Khachai D., Sadykov R., Battaïa O., Khachay M. Precedence constrained generalized traveling salesman problem: Polyhedral study, formulations, and branch-and-cut algorithm. *European J. Oper. Res.*, 2023. Vol. 309, No. 2. P. 488–505. DOI: 10.1016/j.ejor.2023.01.039
11. Khachai D., Battaïa O., Petunin A., Khachay M. Discrete cutting path problems: a general solution framework and industrial applications. *Int. J. Production Res.*, 2025. Vol. 63, No. 3. P. 949–969. DOI: 10.1080/00207543.2024.2365360
12. Moore E. F., Shannon C. E. Reliable circuits using less reliable relays. *J. Franklin Inst.*, 1956. Vol. 262, No. 3. P. 191–208. DOI: 10.1016/0016-0032(56)90559-2
13. Ogorodnikov Y., Rudakov R., Khachai D., Khachay M. A problem-specific branch-and-bound algorithm for the protected shortest simple path problem with must-pass nodes. *IFAC-PapersOnLine*, 2022. Vol. 55, No. 10. P. 572–577. DOI: 10.1016/j.ifacol.2022.09.455
14. Ogorodnikov Yu. Yu., Rudakov R. A., Khachai D. M., Khachai. M. Yu. Fault-tolerant families of production plans: Mathematical model, computational complexity, and branch-and-bound algorithms. *Comput. Math. Math. Phys.* , 2024. Vol. 64, No. 6. P. 1193–1210. DOI: 10.1134/S0965542524700441
15. Rudakov R., Khachay D., Ogorodnikov Y., Khachay M. Reliable production process design problem: Compact MILP model and ALNS-based primal heuristic. In: *Lect. Notes in Comput. Sci., vol. 14395: Optimization and Applications (OPTIMA 2023), N. Olenev et al. (eds)*. Cham: Springer, 2023. P. 174–188. DOI: 10.1007/978-3-031-47859-8_13
16. Rudakov R. A., Khachay D. M., Ogorodnikov Y. Y., Khachay M. Y. Branch-and-price algorithm for the efficient 2-terminal reliability problem. *Sib. Electron. Math. Rep.*, 2025. Vol. 22, No. 2. P. 68–84.
17. Satyanarayana A., Kevin Wood R. A linear-time algorithm for computing k-terminal reliability in series-parallel networks. *SIAM J. Comput.*, 1985. Vol. 14, No. 4. P. 818–832. DOI: 10.1137/0214057
18. Song Y., Luedtke J. R. Branch-and-cut approaches for chance-constrained formulations of reliable network design problems. *Math. Program. Comput.*, 2013. Vol. 5, No. 4. P. 397–432. DOI: 10.1007/s12532-013-0058-3
19. Uchoa E., Pessoa A., Moreno L. *Optimizing with Column Generation: Advanced branch-cut-and-price algorithms (Part I)*. Technical Report L-2024-3. Niterói: Universidade Federal Fluminense, Engenharia de Produção. 2024.
20. Valiant L. G. The complexity of enumeration and reliability problems. *SIAM J. Comput.*, 1979. Vol. 8, No. 3. P. 410–421. DOI: 10.1137/0208032
21. Zenklusen R., Laumanns M. High-confidence estimation of small s-t reliabilities in directed acyclic networks. *Networks*, 2011. Vol. 57, No. 4. P. 376–388. DOI: 10.1002/net.20412