Reliability Engineering and System Safety 248 (2024) 110179

Contents lists available at ScienceDirect

Reliability Engineering and System Safety

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

Time-reliability optimization for the stochastic traveling salesman problem

Wei-Chang Yeh

Integration and Collaboration Laboratory, Department of Industrial Engineering and Engineering Management, National Tsing Hua University, Taiwan

ARTICLE INFO

Keywords:
Traveling Salesman Problem (TSP)
Stochastic TSP (STSP)
Time-reliability optimization
Binary-addition-tree algorithm (BAT)
Pareto solutions

ABSTRACT

This paper presents a novel approach to addressing the Stochastic Traveling Salesman Problem (STSP), a classical problem in combinatorial optimization, by integrating travel time and reliability factors into the decision-making process. Traditional TSP models primarily focus on minimizing the total travel distance or cost without considering the reliability of each route. In real-world situations, especially in logistics and network design, it's just as important to have reliable routes. A reliable route means there's a good chance it will be completed successfully and on time. Our research extends the conventional STSP framework by incorporating a reliability metric for each route, alongside the standard travel time metric. A tri-objective optimization model is proposed to minimize the mean and standard deviation of travel time and maximize route reliability simultaneously. A new algorithm called Permutation Binary-Addition-Tree (BAT) is proposed to solve the problem more efficiently when there is uncertainty. Our approach marks a significant step towards more realistic and practical solutions for route optimization problems in dynamic and uncertain environments. We also present a complexity analysis of our model against traditional cost-only TSP solutions, demonstrating the efficacy of considering reliability in route planning.

1. Introduction

The Traveling Salesman Problem (TSP) is a cornerstone of operations research and combinatorial optimization. It presents both a complex theoretical challenge with significant computational difficulties and a practical puzzle with extensive applications in logistics, routing, urban planning, network design, and transportation. This well-established problem centers on finding the most efficient route to visit a specified set of locations and return to the starting point, traditionally focusing on minimizing measurable factors such as travel time, distance, or cost.

However, the conventional deterministic framework of the TSP, with its emphasis on reducing travel time, distance, or cost, often falls short in capturing the unpredictability and variability inherent in real-life situations, particularly in the dynamic environments of modern logistics and transportation networks. This discrepancy highlights the need for a more nuanced approach that can accommodate the intricacies and ambiguities characteristic of contemporary route optimization challenges.

To bridge this gap, the Stochastic TSP (STSP) has emerged as a promising solution, introducing probabilistic elements into the classical TSP to model uncertainties such as variable travel times and costs. However, despite this significant advancement, a critical aspect often remains overlooked: the reliability of the routes. In real-world scenarios, the probability that a route can be completed successfully and on time, known as its reliability, is just as crucial as the cost incurred. To address this shortcoming, this paper introduces a novel paradigm in the realm of the STSP, emphasizing not only the minimization of the mean and standard deviation of travel time but also the maximization of route reliability under stochastic conditions.

Incorporating reliability into the STSP framework is not merely theoretical; it addresses the urgent needs of various industries. For instance, in logistics and supply chain management, ensuring timely deliveries despite uncertainties like traffic congestion and adverse weather is essential for maintaining service quality and customer satisfaction. Similarly, in telecommunications, reliable data packet routing in unpredictable networks is crucial for service integrity. Thus, our work is motivated by the need to bridge the STSP's theoretical aspects with the practical exigencies of real-world applications, where travel time's efficiency and reliability are intertwined objectives.

The proposed approach, which we term the Time-Reliability STSP (TR-STSP), aims to provide a comprehensive framework for optimizing circular routes in stochastic environments. By considering the probabilistic nature of travel times and the reliability of each route segment, the TR-STSP enables decision-makers to identify routes that not only minimize expected costs but also maximize the likelihood of successful completion within a given time frame. This holistic approach to route

E-mail address: yeh@ieee.org.

https://doi.org/10.1016/j.ress.2024.110179
Received 9 March 2024; Received in revised form 5 April 2024; Accepted 1 May 2024
Available online 6 May 2024
0951-8320/

optimization has the potential to significantly improve the robustness and effectiveness of solutions in various application domains.

In optimization, the significance of travel time and reliability cannot be overstated, as they are pivotal factors that directly influence the efficiency and effectiveness of solutions, particularly in logistics and operational planning, where balancing financial constraints with dependable performance is crucial. This paper presents the development of a tri-objective optimization model for the STSP, treating the mean and standard deviation of travel time and the reliability of routes as three equally important objective functions. The tri-objective optimization model introduces a tri-focused approach, simultaneously addressing travel time, reliability, and a third objective within a single framework, thus offering a more comprehensive and realistic method for tackling complex problems where balancing multiple competing objectives is essential.

We examine the theoretical foundations of this model by extending the Binary Addition Tree (BAT) algorithm to what we call the Permutation BAT, effectively solving the problem. The BAT is an implicit enumeration method designed to completely identify all solutions. The Permutation BAT enhances the efficiency and quality of solutions, particularly in identifying Pareto solutions for the problem. This approach leverages the comprehensive solution-finding capability and efficiency of the BAT, making it well-suited for complex optimization tasks. The goal of our exploration with the Permutation BAT is to provide a more thorough and applicable Pareto solution framework for the STSP, tailored to address the dynamic and uncertain characteristics of real-world routing challenges.

In the subsequent sections, we will review the previous examples of the STSP, multi-objective optimization, and BAT in Section 2. Section 3 delves into the concept of integrating travel time and reliability, outlines the core components, the tri-objective formulation, and provides an example of the problem. Section 4 presents the proposed permutation BAT, including the combination vector, permutation vector, its time complexity, correctness, and an example to demonstrate how to solve the problem with the permutation BAT. Section 5 concludes.

2. Literature review

This section is dedicated to offering a thorough review and analysis of previous research on the STSP, with a particular focus on the latest developments in multi-objective optimization models grounded in Pareto Optimality principles. Additionally, it provides an in-depth overview of the BAT. The objective of this comprehensive overview is not only to present a historical and current perspective on the STSP and tri-objective optimization but also to lay a solid foundation and provide a well-substantiated rationale for the development and implementation of the proposed permutation BAT.

2.1. Previous work on STSP

The TSP is typically formulated to find a circular route that minimizes the total travel distance or time, visiting each city exactly once and returning to the starting city. STSP is an extension of TSP that explores a wide range of methodologies and algorithms to address the problem's inherent stochastic nature effectively. Notable approaches in the research and optimization of STSP include:

1. Probabilistic and Statistical Methods: These approaches involve the use of probability distributions and statistical models to estimate and manage the uncertainties in travel times and costs [1–6]. They are crucial for modeling the stochastic aspects of the problem [7–10].
2. Evolutionary Computing: A class of algorithms, such as genetic algorithms [11,12], that mimic the process of natural selection. They are used in STSP to find optimal routes by generating, evaluating, and evolving a population of solutions over several iterations [11–13].
3. Swarm Intelligence: Swarm intelligence is a collective behavior exhibited by a group of simple, often decentralized agents or organisms that interact with each other and their environment to solve complex problems or achieve a common goal [14–16]. For example, Ant colony optimization inspired by the behavior of ants in finding shortest paths to food sources in exploring and exploiting solutions through pheromone trails in STSP [15]; Particle swarm optimization is a computational method that optimizes a problem by iteratively trying to improve the position and velocity of solutions in STSP for its efficiency in handling complex, multi-dimensional spaces [16].
4. Heuristic Methods: Heuristic methods such as Tabu Search are able to solve STSP [17]. Tabu Search is an iterative search method that navigates the solution space effectively by systematically exploring various feasible solutions while using the tabu list to avoid redundant paths and local optima traps [18]. In STSP, Tabu Search enables the algorithm to explore a broader range of potential solutions and increases the chances of finding a near-optimal solution in a reasonable amount of computational time [17].
5. Neural Networks and Deep Learning: Advanced machine learning techniques, including neural networks [19] and deep learning models [20–26], have been applied to STSP to learn optimal routing patterns and predictions based on historical data [27].
6. Mathematical Programming: Mathematical optimization methods such as linear programming [28,29] and integer programming [30,31] are not typically the primary methods for solving the STSP due to its inherent uncertainty and variability in parameters. These methods are generally more applicable to deterministic problems. However, adaptations or hybrid approaches might be employed in specific STSP scenarios. Dynamic programming can be utilized for certain formulations of STSP [29,30], particularly where the problem can be decomposed into stages with probabilistic outcomes. Its effectiveness, however, is limited by the size and stochastic complexity of the problem.
7. Simulations: Simulation is a probabilistic technique for approximating the global optimum of a given function [32,33]. Monte Carlo simulations [32] and simulated annealing are famous simulations [33]. The former uses repeated random sampling to obtain numerical results, often employed in STSP to evaluate the impact of random variables on the travel paths. The latter is used to find the shortest path by simulating the process of heating and controlled cooling.

Each of these methodologies and algorithms offers unique advantages and can be selected based on the specific characteristics and requirements of the problem at hand, such as the level of uncertainty, size of the problem, and computational resources available [1–33].

2.2. Introduction to multi-objective optimization

Tri-objective optimization, an integral facet of multi-objective optimization, focuses on the simultaneous optimization of two objectives. This area is a subset of multi-objective optimization, which encompasses the optimization of three or more objectives. Both Tri-objective and general multi-objective optimization share the common goal of balancing and optimizing multiple criteria, but tri-objective optimization specifically deals with the unique challenges and interplays between three objectives.

Central to both tri-objective and multi-objective optimization, Pareto Optimality is a criterion where a solution is considered optimal if no objective can be improved without worsening at least one other objective. Based on whether to identify the set of all related Pareto optimal solutions, collectively forming the Pareto front or just a single solution in the Pareto front, these can be categorized as:

Central to both tri-objective and multi-objective optimization, Pareto Optimality is a critical criterion used to evaluate solutions. A solution is deemed Pareto optimal if no objective can be improved without causing a deterioration in at least one other objective. The approaches to identifying optimal solutions within the framework of Pareto Optimality can be broadly categorized based on their focus: either on identifying the entire set of Pareto optimal solutions (forming the Pareto front) or on pinpointing a single optimal solution within the Pareto front. These categories can be described as follows:

1. Pareto Optimality: Central to both bi-objective and multi-objective optimization, Pareto Optimality is a criterion where a solution is considered optimal if no objective can be improved without worsening at least one other objective. The function values of the pareto solution are typically referred to as Pareto front. The aim is to identify the set of all Pareto optimal solutions, collectively forming the Pareto front. These can be categorized as:
   - Exact Pareto front: This represents the true set of Pareto optimal solutions, ideally covering all possible trade-offs between objectives. Identifying the exact Pareto front is often challenging, especially in complex or high-dimensional problems. For example, the BAT proposed in [34].
   - Approximated Pareto front: Advances in computational techniques, particularly the integration of machine learning with multi-objective optimization, have enhanced the ability to approximate the Pareto front in complex scenarios. These methods, including algorithms like NSGA-II (Non-dominated Sorting Genetic Algorithm II) [22], MOPSO (Multi-Objective Particle Swarm Optimization) [35], MSSO (Multi-Objective SSO) [36], or SPEA2 (Strength Pareto Evolutionary Algorithm 2) [37], are adept at efficiently exploring a diverse set of solutions that approximate the Pareto front in multi-objective optimization problems.
2. Single-Solution Techniques: These techniques focus on deriving a single, optimal solution per run in a multi-objective optimization problem. They can be classified into two main approaches:
   - Conversion to Single-Objective: Methods like the weighted sum method [38] transform a multi-objective problem into a single-objective one by assigning weights to each objective, thereby simplifying the optimization process.
   - Focused Pareto Solution: Other techniques yield only one solution on the Pareto front based on decision-makers' preferences, representing a specific trade-off among objectives. This category includes the ε-constraint method [39], Analytic Hierarchy Process (AHP) [40], Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS) [41], interactive methods [42], and goal programming [43]. These methods are guided by specific criteria or decision-makers' preferences to identify an optimal solution that best fits the given constraints or goals.

In summary, Pareto Optimality is crucial in multi-objective optimization, encompassing bi-objective scenarios [35–37]. It guides the development of methods that approximate the Pareto front or yield single solutions, adapting to the complexity and specificities of objective interplays. While tri-objective optimization is a subset of multi-objective optimization, the methodologies employed often overlap, with the primary distinction being the scope and depth of objective analysis [38–43]. These approaches collectively offer a robust framework for addressing complex optimization challenges across various domains.

2.3. Overview of BAT

Yeh [34] developed the BAT to efficiently produce all possible combinations of these m-tuple binary-state vectors using the binary addition principle, encompassing both feasible and infeasible combinations. The BAT algorithm is characterized by its use of a binary tree structure for efficient hierarchical organization, a bottom-up approach for effective carry handling, parallel processing capabilities for enhanced speed [44], and scalability for managing binary operations of varying lengths [45,46].

The BAT offers several significant advantages. The approach is both straightforward and direct, making it very efficient, especially for large-scale binary operations. As a result, it significantly reduces computational time [47]. This efficiency is further enhanced by its capability for parallel processing, which is a crucial feature in speeding up binary calculations [48]. Additionally, the BAT algorithm ensures accuracy and reliability in its results, an essential aspect for any computational task where precision is critical [34]. Its scalability to handle binary numbers of various lengths also broadens its applicability across different domains, from basic computing tasks to more complex applications like the propagation of wildfire [49] and computer viruses [50].

Despite its straightforward approach, the BAT does have certain limitations [46,47]. While efficient for managing binary addition, especially in large-scale computations, its simplicity might not be well-suited for more complex operations that require advanced algorithms with more sophisticated handling capabilities [46]. Additionally, in scenarios where nuanced or specialized processing of binary data is required, the BAT algorithm's direct methodology might fall short in terms of flexibility and adaptability [47]. This makes it less ideal for tasks that demand intricate manipulation or analysis of binary data beyond basic addition.

BAT emulates incrementing a binary number, starting from the least significant bit and progressing to the most significant bit in the binary-state vector, as described in Ref. [34]. The fundamental steps of the BAT are outlined in the following pseudocode [34], with its full implementation available in [44]:

Procedure BAT

Input: m.
Output: All m-tuple binary-state vectors.
STEP 1. Initialize $i = 1$ and set Z as an m-tuple vector of zeros.
STEP 2. If $Z(a_i) = 0$, change $Z(a_i)$ to 1, reset i to 1, generate a new Z, and repeat STEP 2.
STEP 3. If $i < m$, set $X(a_i)$ to 0, increment i, and go back to STEP 2. If not, terminate the algorithm.

This pseudocode illustrates BAT's simplicity, adaptability, and efficiency in both computational time and memory usage. Comparative studies [34,44,46] have demonstrated BAT's superior performance over other recognized search methods like Depth-First Search [52–56], Breadth-First Search [34], and universal generating function method [51]. As a result, various adaptations and applications of BAT have been developed [34,44,46].

For instance, consider the network with four nodes: 1, 2, 3, 4, and six arcs, as shown in Fig. 1. There is a total of $2^4 = 16$ combinations for these four nodes, from $Z_1 = (0, 0, 0, 0)$ to $Z_{16} = (1, 1, 1, 1)$, for the execution of the BAT, as detailed in Table 1.

2.4. STSP example without considering the probability

In the context of the STSP, the combination of the expected travel time and its standard deviation provides a strategic approach for navigating uncertain environments. The expected travel time represents the average duration of travel, while the standard deviation indicates the variability of that travel time. By considering both of these factors,

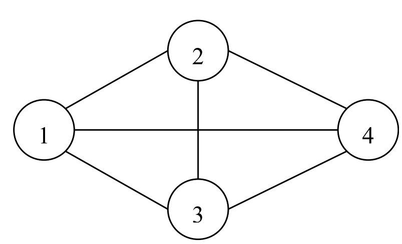

Fig. 1. The bridge network.

Table 1
All combinations of $Z_i = (z_{i,1}, z_{i,2}, z_{i,3}, z_{i,4})$.

i	$z_{i,1}$	$z_{i,2}$	$z_{i,3}$	$z_{i,4}$	i	$z_{i,1}$	$z_{i,2}$	$z_{i,3}$	$z_{i,4}$
1	0	0	0	0	9	0	0	0	1
2	1	0	0	0	10	1	0	0	1
3	0	1	0	0	11	0	1	0	1
4	1	1	0	0	12	1	1	0	1
5	0	0	1	0	13	0	0	1	1
6	1	0	1	0	14	1	0	1	1
7	0	1	1	0	15	0	1	1	1
8	1	1	1	0	16	1	1	1	1

decision-makers can make more informed and predictability-focused decisions in routing and travel planning, effectively balancing efficiency with the need to manage uncertainty.

In the simplified example of the STSP with four cities labeled 1, 2, 3, and 4, as depicted in Fig. 1, Table 2 provides the mean travel time (in hours) and its standard deviation (also in hours) for each pair of distinct cities. The travel time between two nodes can differ due to directional traffic patterns, road layout and infrastructure differences, geographical and environmental factors, regulatory constraints, construction activities, and natural phenomena, making route optimization reflect real-world complexities.

For the simplified example of the STSP with 4 cities (1, 2, 3, 4) as shown in Fig. 1. Table 2 presents the mean travel time and its standard deviation for each pair of distinct cities.

For example, the mean travel time and standard deviation for the arc connecting cities 2 to 4 are represented as 6.55 h and 1.08 h, respectively, denoted by $E(T(e_{1,3})) = 6.55$ and $\sigma(T(e_{1,3})) = 1.08$.

Given a small example with 4 cities, it is practical to enumerate all possible routes. Let $X_i = (x_{i,1}, x_{i,2}, x_{i,3}, x_{i,4})$ be the i^{th} circular route from nodes $x_{i,1}$ to $x_{i,2}$ to $x_{i,3}$ to $x_{i,4}$ and back to $x_{i,4}$. For 4 cities, taking one city as the starting and ending point and permuting the remaining 3 cities gives us $3! = 6$ possible circular routes, i.e., X_i for $i = 1, 2, \ldots, 6$. This approach allows us to systematically evaluate and calculate the total expected travel times and their standard deviations for each circular route, as presented in Table 3.

For instance, the total travel time for the circular route 1 -> 2 -> 4 -> 3 -> 1 is 24.34 h, with a standard deviation of 2.09692 h. The breakdown is as follows: from city 1 to city 2, it takes 8.33 h with a standard deviation of 1.32 h; from city 2 to city 4, it takes 5.01 h; from city 4 to city 3, it takes 4.06 h; and from city 3 back to city 1, it takes 8.69 h.

The aim is to identify a circular route that minimizes the total expected travel time, ensuring that each city is visited only once before returning to the start, despite variable and probabilistic travel times. To manage this variability, strategies include opting for circular routes with the lowest expected travel time, the least variability (standard deviation), or the best balance between these criteria. Details are as follows:

1. Lowest Expected Travel Time

The focus is on finding the circular route with the lowest total average travel time between city pairs. The circular route from city 1 to 4 to 2 to 2 and back to 1 has the lowest expected travel time of 22.91 h, as shown in the "travel time" column of Table 3.

1. Least Variability

This approach targets the circular route with the lowest overall standard deviation in travel times to reduce uncertainty. According to the "standard deviation" column of Table 2, the circular route 1 -> 3 -> 2 -> 4 has the least variability, with a total standard deviation of 1.26764 h.

Table 2
Mean travel time and standard deviation for Fig. 1.

		1	2	3	4
average	1		8.33	9.46	4.45
	2	9.01		5.01	6.55
	3	8.69	5.39		5.43
	4	5.57	4.98	4.06	
standard deviation	1		1.32	0.62	1.43
	2	0.88		1.06	1.08
	3	0.27	0.49		0.49
	4	0.45	0.98	1.01	

3. The proposed time-reliability RTSP and example

The TR-STSP emerges as a seminal advancement within the domain of combinatorial optimization, meticulously integrating the unpredictable nature of travel parameters with the critical dimension of path reliability. This enhancement significantly enriches the conventional STSP framework, rendering it adept at navigating the intricate dynamics and uncertainties pervasive in real-world logistical contexts.

3.1. Augmenting the STSP with reliability and core components of the TR-STSP

The TR-STSP introduces a paradigmatic shift by embedding a reliability dimension within the stochastic modeling of travel parameters. This strategic inclusion extends the model's capacity to account for a wide array of contingencies—ranging from environmental disruptions to socio-political upheavals—that could impinge on the availability or functionality of predefined circular routes. The probabilistic assessment of arc functionality stands as a testament to the model's responsiveness to fluctuating operational landscapes, ensuring that routing solutions not only excel in efficiency but are also imbued with resilience and dependability.

This evolution from a theoretical construct to a pragmatic toolkit encapsulates a holistic circular route optimization methodology. It balances speed, cost, and reliability, thereby mitigating risks associated with operational delays and their ensuing costs. In doing so, it addresses the exigencies of sectors where the assurance of circular route usability is paramount, including the transportation of hazardous materials and the maintenance of uninterrupted services in critical infrastructure networks.

The TR-STSP's architecture is underpinned by the symbiotic relationship between stochastic variables and the reliability function. This dual framework not only captures the inherent variability of travel parameters through a probabilistic lens but also rigorously evaluates the operational viability of circular route segments. The stochastic modeling component acknowledges that travel times, distances, and costs are not immutable but are influenced by a plethora of external factors. Concurrently, the reliability function provides a mechanism for assessing the likelihood that each arc remains viable, incorporating environmental, socio-political, and infrastructural considerations into the operational continuity evaluation.

By amalgamating stochastic analysis with reliability assessment, the TR-STSP articulates a comprehensive circular route optimization framework. This approach significantly elevates the precision of routing models and the depth of strategic decision-making, enabling the formulation of routing strategies that are not merely efficient but are resilient to disruptions and consonant with overarching reliability and service continuity objectives.

3.2. Tri-objective formulation of the TR-STSP

The TR-STSP distinguishes itself through a tri-objective formulation that aspires to concurrently minimize expected travel metrics, maximize circular route reliability, and embed risk management within the optimization process.

Table 3
All possible circular routes and their information.

i	$x_{i,1}$	$x_{i,2}$	$x_{i,3}$	$x_{i,4}$	travel time	standard deviation	$z_{i,1}$	$z_{i,2}$	$z_{i,3}$	$z_{i,4}$
1	1	2	3	4	24.34	2.09692	0	0	0	0
2	1	2	4	3	27.63	2.33289	0	0	0	1
3	1	4	3	2	**22.91**	2.42478	0	0	0	2
4	1	3	2	4	26.97	**1.26764**	0	0	1	0
5	1	3	4	2	28.88	1.31433	0	0	1	1
6	1	4	2	3	23.13	2.52424	0	0	1	2

Let $X = (x_1, x_2, \ldots, x_n)$ be a permutation of all nodes in $V = \{1, 2, \ldots, n\}$ representing the circular route from nodes $x_1 = 1$ to x_2 to … to x_n and back to $x_1 = (n+1)$. Let $T(e_{i,j})$, $\sigma(e_{i,j})$, and $R(e_{i,j})$ be the expected travel time, standard deviation, and reliability of the arc from nodes i to j, respectively. The tri-objective formulation of the TR-STSP is listed below:

$$\text{Min } T(X) = \sum_{i=1}^{n} T\left(e_{x_i, x_{(i+1)}}\right) \tag{1}$$

$$\text{Min } \sigma(X) = \sqrt{\sum_{i=1}^{n} \left[\sigma\left(e_{x, x_{(i+1)}}\right)\right]^2} \tag{2}$$

$$\text{Max } R(X) = \prod_{i=1}^{n} R\left(e_{x_i, x_{(i+1)}}\right) \tag{3}$$

To simplify the above calculations, Eqs. (2) and (3) are reduced to:

$$\text{Min } \sigma^*(X) = \sum_{i=1}^{n} \left[\sigma\left(e_{x_i, x_{(i+1)}}\right)\right]^2 \tag{4}$$

$$\text{Max } \mathrm{R}^*(X) = \sum_{i=1}^{n} \log\left(R\left(e_{a_i, a_{(i+1)}}\right)\right) \tag{5}$$

This triadic ambition necessitates a departure from conventional optimization techniques towards more intricate, multi-dimensional analytical methods. The integration of a probabilistic assessment of travel variables with a qualitative evaluation of circular route viability challenges traditional optimization paradigms, necessitating advanced algorithmic strategies to reconcile these competing objectives.

This holistic approach is indispensable across various sectors, including logistics and emergency response, where the mitigation of circular route unreliability is critical. It equips stakeholders with a robust and adaptable framework, ensuring that routing strategies are not only efficient under nominal conditions but are also resilient to operational uncertainties.

The TR-STSP's methodological innovations contribute profoundly to the theoretical and practical advancement of routing optimization. It underscores a shift towards more adaptive, risk-informed logistics and transportation planning strategies, emphasizing the integration of sophisticated risk assessment and management techniques. Through its nuanced balance of efficiency, reliability, and risk mitigation, the TR-STSP encapsulates a forward-thinking paradigm in the pursuit of sustainable and dependable routing solutions amidst the complexities of the modern logistical landscape.

3.3. Example for TR-STSP

Incorporating arc reliability into the routing problem significantly alters the optimization criteria, transitioning from a sole focus on minimizing expected travel time and standard deviation to also maximizing the probability of successfully completing the trip via operational arcs. This adjustment introduces a tri-dimensional optimization problem that encompasses the physical, economic, and now reliability dimensions of routing, providing a holistic approach to address the dynamic uncertainties of logistics and transportation planning.

Table 4, detailing the operational probabilities for each arc in Fig. 1, exemplifies this shift by quantifying the likelihood of arc functionality, such as a 0.93 reliability score for the arc from cities 1 to 2 indicating an 93 % probability of operability. This data is pivotal for identifying circular routes that are not only efficient and stable in terms of travel time and variability but also reliable and likely to be uninterrupted by non-functional arcs.

As mentioned in Section 2.4, there are $3! = 6$ circular routes in Fig. 1. The reliability of each circular route is listed in Table 5. For example, based on Eq. (3), the reliability for the circular route $X_2 = (1, 2, 4, 3)$, i.e., 1 -> 2 -> 3 -> 4 -> 1 is calculated as

$$0.93 \times 0.80 \times 0.83 \times 0.88 = 0.54342. \tag{6}$$

Alternatively, if the logarithm of each arc's reliability is calculated in advance, we can use Eq. (5):

$$\begin{aligned}&\exp(\ln(0.93) + \ln(0.80) + \ln(0.83) + \ln(0.88)) = \exp(-0.609877194)\\ &= 0.54342.\end{aligned} \tag{7}$$

This comprehensive approach underscores the necessity of evaluating not just the direct aspects of circular route selection but also the probabilistic reliability of each path, ensuring strategies are robust against potential disruptions. However, it is also acknowledged that for larger networks, enumerating and evaluating all possible circular routes becomes computationally impractical due to the factorial explosion in the number of circular routes. In such scenarios, sophisticated heuristic algorithms are employed to identify pareto-optimal solutions, balancing efficiency, variance, and reliability in a computationally feasible manner, thereby adapting to the complex and uncertain nature of real-world transportation networks.

4. Proposed permutation BAT

In this section, we present a new method called permutation BAT. It creates all different arrangements of a set without repeating any of them. This method addresses and corrects the limitations observed in the multi-state BAT [34], which incorrectly produces h^h combinations instead of the correct $h!$ permutations for a set consisting of h elements.

4.1. The foundational concept

The multi-state BAT extended from the BAT generates all combinations of m-tuple vector $Z = (z_1, z_2, \ldots, z_n)$ such that $z_i = \{0, 1, \ldots, \mathrm{UB}(z_i)\}$ and $\mathrm{UB}(z_i)$ is the upper-bound of z_i for $i = 1, 2, \ldots, n$.

The multi-state BAT can obtain all permutations. However, it requires a filtration process to extract all permutations from the total set of combinations. For example, consider $Z_i = (z_{i,1}, z_{i,2}, z_{i,3})$. Table 6

Table 4
Operational probability of each arc in Fig. 1.

	1	2	3	4
1		0.93	0.98	0.90
2	0.89		0.80	0.80
3	0.86	0.82		0.83
4	0.88	0.93	0.91	

Table 5
The reliability of each circular route.

	z_0	z_1	z_2	z_3	reliability
1	1	2	3	4	0.54342
2	1	2	4	3	0.58225
3	1	4	3	2	0.59771
4	1	3	2	4	0.56573
5	1	3	4	2	0.67325
6	1	4	2	3	0.57586

Table 6
Enumeration of all 27 combinations of the set {0, 1, 2}.

i	X_i			i	X_i			i	X_i		
1	0	0	0	10	1	0	0	19	2	0	0
2	0	0	1	11	1	0	1	20	2	0	1
3	0	0	2	**12**	**1**	**0**	**2**	21	2	0	2
4	0	1	0	13	1	1	0	**22**	**2**	**1**	**0**
5	0	1	1	14	1	1	1	23	2	1	1
6	**0**	**1**	**2**	15	1	1	2	24	2	1	2
7	0	2	0	**16**	**1**	**2**	**0**	25	2	2	0
8	**0**	**2**	**1**	17	1	2	1	26	2	2	1
9	0	2	2	18	1	2	2	27	2	2	2

enumerates all $3^3 = 27$ combinations for Z_i, where $z_{i,k} = 0, 1$, $\text{UB}(z_{i,k}) = 2$ for $k = 1, 2, 3$. However, out of these 27 combinations, only Z_6, Z_8, Z_{12}, Z_{16}, and Z_{22} (denoted in bold) represent the correct permutation of the set {0, 1, 2}.

Therefore, there is a need to address and refine the limitations observed in the multi-state BAT to generate all permutations without the need to filter from all combinations, as demonstrated above.

4.2. Setting up the upper bound For each coordinate in multi-state BAT

The initial step in the proposed permutation BAT involves setting the upper bound for each coordinate. This is accomplished by setting $\text{UB}(z_{i,k}) = (k-1)$, i.e., $z_{i,k} = 0, 1, \ldots, (k-1)$ for all $k = 1, 2, \ldots, n$. For example, $z_{i,5}$ can only be an integer within the set {0, 1, 2, 3, 4}. With this setup, the total number of combinations in the multi-state BAT is reduced from n^n to $n!$, aligning precisely with the number of permutations of the set $\{0, 1, \ldots, (n-1)\}$. For instance, the 27 combinations of the set {0, 1, 2} listed in Table 6 are pared down to 6, as illustrated in the columns under Z_i in Table 7.

However, not every Z_i in Table 7 constitutes a permutation, which necessitates that each element in the set appears exactly once. For example, all coordinates in Z_1 are zero. To address this issue, a new vector, named the permutation vector, is proposed, as shown in the columns under X_i in Table 7. This vector is designed to generate the corresponding permutation of the combinations based on the results obtained from the multi-state BAT after establishing the upper bound for each coordinate.

4.3. Permutation vector

Each vector obtained in the BAT and the multi-state BAT is referred to as a combination vector. Given a permutation vector $X_i = (x_{i,1}, x_{i,2}, \ldots, x_{i,h})$, we initialize the permutation vector $X_i = (0, 1, \ldots, (n-1))$. Upon identifying the j^{th} non-zero integer, say $x_{i,k} = \alpha > 0$, in X_i, the k^{th} coordinate in the current permutation vector is shifted α positions earlier, resulting in an updated permutation vector. This process always starts with the first non-zero integer, then proceeds to the next non-zero integer, and continues in this sequence until the last non-zero integer is processed.

This method ensures that each element in the set appears exactly once, thereby generating the correct permutation corresponding to the combination represented by X_i. For instance, consider $Z_2 = (0, 0, 1)$ in Table 7. Since $z_{2,3} = 1$ is the first and also the only one non-zero coordinate in Z_2, the third coordinate in the initial permutation vector (0, 1, 2) is 2. The result, after shifting the third coordinate one position earlier, is $X_3 = (0, 2, 1)$.

Taking $Z_6 = (0, 1, 2)$ from Table 7 as another example, since $x_{6,2} = 1$ is the first non-zero coordinate in X_6, the second coordinate in the initial permutation vector $X_6 = (0, 1, 2)$ is 1. The updated permutation vector, after shifting the second coordinate one position earlier, is $X_6 = (1, 0, 2)$. Next, $z_{6,3} = 2$ is the second and also the last non-zero coordinate in Z_6; hence, the third coordinate in the current permutation vector X_6 is 2. The final updated permutation vector, after shifting the third coordinate two positions earlier, is $X_6 = (2, 1, 0)$.

In the same way, we have all permutation X_i as shown in the last part of Table 6, for combination Z_i listed in the second part of Table 7.

4.4. Pseudocode and time complexity

In the TSP, the objective is to find the shortest possible route that visits each city once and returns to the starting city. The route can be represented as a circular permutation, where the starting and ending cities are the same. For simplicity, we can assume that the starting city is always node 0. This assumption does not affect the generality of the problem, as any circular permutation starting and ending at a different city can be rotated to start and end at node 0. For example, the permutation 0, 1, 2, …, $(n-1)$, 0 is equivalent to 1, 2, …, $(n-1)$, 0, 1. Therefore, we can focus on finding the optimal permutation for the remaining cities, which are nodes 1, 2, …, $(n-1)$, with the understanding that the route starts and ends at node 0.

Based on Section 4.3 and the above paragraph, the following pseudocode outlines the process for updating a permutation vector X_i from $(1, 2, \ldots, (n-1))$ to $(x_{i,1}, x_{i,2}, \ldots, x_{i,(n-1)})$ based on a given combination vector $Z_i = (z_{i,1}, z_{i,2}, \ldots, z_{i,(n-1)})$, where $z_{i,k} = 0, 1, \ldots, (k-1)$ for all $k = 1, 2, \ldots, (n-1)$, ensuring that each element in the set $\{1, 2, \ldots, (n-1)\}$ appears exactly once.

Algorithm: Update Permutation Vector

Input: The combination vector $Z_i = (z_{i,1}, z_{i,2}, \ldots, z_{i,(n-1)})$.
Output: Updated permutation vector $X_i = (x_{i,1}, x_{i,2}, \ldots, x_{i,(n-1)})$.
STEP 0. Initialize the permutation vector X_i with elements 1, 2, …, $(n-1)$ and let N be the set of indices of non-zero integers in Z_i, sorted from smallest to largest.
STEP 1. If $N = \varnothing$, i.e., Z_i is a n-tuble zero vector, then $X_i = (1, 2, \ldots, (n-1))$ is the permutation corresponding to Z_i, and the process halts.
STEP 2. Let $k = 1$.
STEP 3. Let $\alpha = z_{i,\kappa}$, where κ is the k^{th} element in N
STEP 4. Determine the target position for the shift: $t = (k - \alpha)$.
STEP 5. Update X_i by shifting its κ^{th} element to the target position t.
STEP 6. If $k < |\text{N}|$, then let $k = k + 1$ and go to STEP 2.

The time complexity is $O((n-1))$ for setting up the initial permutation vector X_i in STEP 0, identifying all non-zero integers to obtain N in STEP 1, and shifting the coordinate in STEP 4. Hence, the time complexity of the above procedure for generating the corresponding X_i using the permutation vector method is $O((n-1)|\text{N}|)$.

Because $z_{i,k} = 0, 1, 2, \ldots, (k-1)$ and each digit in $0, 1, \ldots, (n-1)$ appears with the same frequency, i.e., $(n-1)!/k$, there are $(k-1)\times((n-1)!/k)$ non-zero integers $x_{i,k}$ among $(n-1)!$ combinations. Thus, the

Table 7
All 6 combination vectors and their corresponding permutation vectors.

i	Z_i			X_i		
1	0	0	0	0	1	2
2	0	0	1	0	2	1
3	0	0	2	2	0	1
4	0	1	0	1	0	2
5	0	1	1	1	2	0
6	0	1	2	2	1	0

number of non-zero integers in $(n-1)!$ combinations is

$$1\times((n-1)!)/2+2\times((n-1)!)/3+3\times((n-1)!)/4+\ldots.+((n-1)-1)\times((n-1)!)/(n-1)$$

$$\approx((n-1)!)[((n-1)-1)-ln((n-1)-1)-\gamma], \tag{8}$$

where γ is the Euler-Mascheroni constant (approximately 0.57721). However, this approximation becomes less accurate for small values of n [1].

From the above, the time complexity for generating all permutations X_i using the permutation vector method is

$$O((n-1)\times(n-1)!\times[((n-1)-1)-ln((n-1)-1)-\gamma]) \approx O\left((n-1)^2\times(n-1)!\right)\approx O(n\times n!). \tag{9}$$

4.5. Correctness of the proposed permutation BAT

The proposed permutation BAT is considered correct if it generates exactly $(n-1)!$ permutations of the numbers 1, 2, …, $(n-1)$ without any duplications, ensuring both completeness and uniqueness are satisfied.

Because no duplicated combinations are found in the BAT. If there exists a one-to-one relationship between each combination vector Z_i obtained from the multi-state BAT and the related permutation vector X_i obtained from the proposed permutation BAT, then no permutation is duplicated. It is evident that from Z_i, one can find one and only one X_i using the proposed permutation BAT. If this relationship holds true in reverse as well, where Z_i can be found from X_i, then there is a one-to-one correspondence between the combination vector and the permutation vector.

To reverse the procedure outlined in Section 4.4, we define a misplaced coordinate as $x_{i,k}\neq(k-1)$ in the permutation vector X_i. Let Z_i be an $(n-1)$-tuple zero vector. We initiate the process by identifying the smallest misplaced coordinate in X_i, denoted as $\kappa=x_{i,k}\neq(k-1)$, while $x_{i,j}=(j-1)$ for all $j<k$. We update X_i by shifting and inserting its k^{th} element to the target position (κ+1) and set $z_{i,(\kappa+1)}=(k-1)-\kappa$. This process continues until X_i is updated to (0, 1, …, $(n-1)$), resulting in the exact corresponding combination vector Z_i.

For example, $x_{6,1}=2$ is identified as the smallest misplaced coordinate in $X_6=(2, 1, 0)$ because it is not 0. To correct this, we shift the value 2 back to position 3, updating X_6 from (2, 1, 0) to (1, 0, 2), and let $z_{6,3}=2$. Subsequently, $x_{6,1}=1$ becomes the smallest misplaced coordinate in new $X_6=(1, 0, 2)$. X_6 is then updated from (1, 0, 2) to (0, 1, 2), and $z_{6,2}$ is updated from 0 to 1, resulting in $X_6=(0, 1, 2)$ and $Z_6=(0, 1, 2)$.

Through this process, a one-to-one relationship between the combination vector and the permutation vector is established. Consequently, all permutation vectors are obtained from the proposed permutation BAT without any duplicates.

4.6. Example

Consider a logistics company that needs to deliver goods to five different locations within a city. The locations are labeled as 0, 1, 2, 3, and 4. The company has one delivery truck that needs to start from the warehouse (location 0), visit each of the other locations exactly once to deliver goods, and then return to the warehouse.

The travel times between locations are uncertain due to factors like traffic congestion, road conditions, and weather, so they are represented by stochastic variables with known means (MEAN), standard deviations (STDEV), and operational reliabilities (REL) as listed in Table 8. This data can be gathered from historical records, traffic models, or real-time monitoring systems.

The company wants to find the optimal route for the delivery truck that minimizes the expected total travel time while also considering the reliability and variability of the travel times. This is important because a route with a lower expected travel time but high variability might result in late deliveries, while a route with slightly higher expected travel time but lower variability and higher reliability might be more desirable for ensuring on-time deliveries.

Table 8
Information of five cities labeled 0, 1, 2, 3, and 4.

		0	1	2	3	4
MEAN	0		5.19492	4.47737	8.71896	3.08719
	1	6.64223		8.27607	3.63506	3.08963
	2	5.18394	6.49062		7.29069	6.70962
	3	2.26637	3.24760	4.79061		6.70693
	4	3.37162	4.28620	2.42409	2.80642	
STDEV	0		1.10778	0.40632	0.85929	0.07109
	1	1.48274		1.39640	0.73185	1.47569
	2	1.00290	1.31071		0.57483	0.94623
	3	0.67161	1.25971	1.18372		0.66295
	4	0.59108	0.53111	1.27276	0.61127	
REL	0		0.96150	0.93966	0.82268	0.82821
	1	0.88556		0.89510	0.81049	0.87028
	2	0.90979	0.89152		0.94462	0.84089
	3	0.87013	0.81257	0.91653		0.86244

To solve this TR-STSP, the company uses a multi-objective optimization approach to generate a set of Pareto-optimal routes. Each route in this set represents a trade-off between the expected travel time, variability, and reliability. The company then evaluates these routes based on their specific business priorities, such as the importance of on-time deliveries, fuel costs, driver hours, and risk aversion to select the most suitable route from the set of Pareto-optimal routes for the delivery truck.

By using the TR-STSP approach, the logistics company can make more informed decisions that account for the inherent uncertainties in travel times, leading to more efficient and reliable delivery operations. The step-by-step approach for finding the permutation in Pareto front to this TR-STSP using the proposed permutation BAT is listed below and each permutation represents a potential route for the delivery truck.

In STEP 0 of the proposed algorithm, the initial permutation is (1, 2, 3, 4) without considering the start city 0, and it is (0, 1, 2, 3, 4, 0) if considering the start city. Hence, in such a permutation, the mean travel time is 5.19492 + 8.27607 + 7.29069 + 6.70693 + 3.37162 = 31.75481, the standard deviation of travel time is the square root of $1.10778^2+1.39640^2+0.57483^2+0.66295^2+0.59108^2=0.63967$, and the operation reliability is $0.96150\times0.89510\times0.94462\times0.86244\times0.87013=0.63745$. Hence, the tri-objective function value $F(X_1)$ is (mean, standard deviation, reliability) = (31.75481, 0.63967, 0.63745). Because X_1 is the first permutation, $F(X_1)$ = (31.75481, 0.63967, 0.63745) is added to the candidate Pareto front Φ = { $F(X_1)$ = (31.75481, 0.63967, 0.63745)}.

From the multi-state BAT and permutation BAT, the second combination and permutation are $Z_2=(0, 0, 0, 1)$ and $X_2=(1, 2, 4, 3)$, respectively. $F(X_2)$ = (26.23463, 1.86727, 0.55661) is not dominated by any element in Φ = { $F(X_1)$}. Hence, it is also added to Φ, i.e., Φ = { *ν*, $F(X_2)$ = (26.23463, 1.86727, 0.55661)}.

For $i=3$, we have $Z_3=(0, 0, 0, 2)$, $X_3=(1, 4, 3, 2)$, and $F(X_3)$ = (22.37220, 2.24819, 0.58667). Because $F(X_3)$ is not dominated by any element in Φ, $F(X_3)$ is added to Φ. For $i=4$, we have $Z_4=(0, 0, 0, 3)$, $X_4=(4, 2, 3, 1)$, and $F(X_4)$ = (18.80325, 1.99168, 0.72126). Because $F(X_4)$ dominates $F(X_3)$ and is not dominated by the other elements in Φ, Φ is updated from { $F(X_1)$, $F(X_2)$, $F(X_3)$} to { $F(X_1)$, $F(X_2)$, $F(X_4)$}.

Following the proposed algorithm, we identified Pareto front Φ = {F_1, F_2, F_4, F_5, F_{10}, F_{13}, F_{18}, F_{19}} which included all Pareto-optimal routes as shown in Table 9. No route in this set can be improved in one of these aspects without worsening at least one of the other aspects. Decision makers can utilize these solutions to inform their decisions.

The chosen route serves multiple purposes, including the execution

Table 9
The procedure to have complete Pareto front Φ.

	x_1	x_2	x_3	x_4	z_1	z_2	z_3	z_4	MEAN	STDEV	REL	Parto solution
1	0	0	0	0	1	2	3	4	31.75481	**0.63967**	0.63745	Yes
2	0	0	0	1	1	2	4	3	26.23463	1.86727	0.55661	Yes
3	0	0	0	2	1	4	3	2	22.37220	2.24819	0.58667	
4	0	0	0	3	4	2	3	1	**18.80325**	1.99168	**0.72126**	Yes
5	0	0	1	0	1	3	2	4	24.61641	1.84113	0.52796	Yes
6	0	0	1	1	1	3	4	2	24.45161	2.51275	0.54622	
7	0	0	1	2	1	4	2	3	21.24693	2.29162	0.54895	
8	0	0	1	3	4	3	2	1	19.92853	2.60233	0.67222	
9	0	0	2	0	3	2	1	4	27.37602	2.79293	0.60708	
10	0	0	2	1	3	2	4	1	27.25907	1.49019	0.65796	Yes
11	0	0	2	2	3	4	1	2	34.47878	2.61405	0.55034	
12	0	0	2	3	4	2	1	3	18.88455	2.40493	0.55171	
13	0	1	0	0	2	1	3	4	25.59618	1.43720	0.55965	Yes
14	0	1	0	1	2	1	4	3	20.11164	2.98429	0.58944	
15	0	1	0	2	2	4	3	1	19.99469	1.99249	0.58341	
16	0	1	0	3	4	1	3	2	22.28968	2.58989	0.63676	
17	0	1	1	0	2	3	1	4	22.39149	2.80806	0.48444	
18	0	1	1	1	2	3	4	1	25.51488	1.45950	0.59081	Yes
19	0	1	1	2	2	4	1	3	22.35585	1.69842	0.45419	Yes
20	0	1	1	3	4	3	1	2	23.90789	3.13322	0.53345	
21	0	1	2	0	3	1	2	4	31.23844	2.48786	0.60262	
22	0	1	2	1	3	1	4	2	23.97089	3.43856	0.64419	
23	0	1	2	2	3	4	2	1	27.09427	2.62445	0.63760	
24	0	1	2	3	4	1	2	3	26.18775	2.81702	0.66060	

of delivery operations, scheduling the delivery truck, preparing for unexpected events, and sharing the plan with relevant stakeholders. By consistently monitoring the performance of the delivery operations and making adjustments to the route when needed, based on real-time data and evolving conditions, the company can improve both the efficiency and reliability of its delivery services.

5. Conclusions

The proposed TR-STSP model is developed under stochastic conditions, where travel times are subject to uncertainty, and reliability scores are assigned based on various factors such as traffic conditions, weather forecasts, infrastructure issues, and historical data. A novel BAT, called the permutation BAT, is proposed to find the Pareto front of the TR-STSP. By incorporating both stochastic elements and the operational viability of each connection, the TR-STSP offers a robust solution to the multifaceted challenges of routing optimization in uncertain environments.

In this study, we have proposed a novel TR-STSP, which addresses the intricacies of routing optimization under stochastic conditions. The model incorporates the uncertainty of travel times and the reliability of routes, providing a comprehensive framework for logistics planning in environments characterized by unpredictability.

To solve the TR-STSP, we introduced a new BAT, termed the permutation BAT, specifically tailored to tackle the challenges of stochastic optimization. This algorithm efficiently navigates the solution space to identify the Pareto front, representing the optimal trade-offs between travel time, reliability, and other pertinent criteria.

The implementation of the TR-STSP model and the permutation BAT algorithm offers substantial benefits for logistics and transportation planning. By integrating stochastic elements and operational viability into the decision-making process, these tools enable organizations to develop routing strategies that are both robust and resilient. This is particularly vital in the current dynamic and uncertain landscape, where the capacity to adapt to fluctuating conditions and mitigate risks is crucial for maintaining operational efficiency and reliability.

Future research could focus on further refinement of the TR-STSP model and the permutation BAT algorithm, extending their applicability to a broader range of scenarios, and incorporating additional factors such as cost, environmental impact, and social considerations into the optimization process. By advancing our understanding and capabilities in this area, we can contribute to the development of more effective and sustainable logistics and transportation systems.

CRediT authorship contribution statement

Wei-Chang Yeh: Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Software, Resources, Project administration, Methodology, Investigation, Funding acquisition, Formal analysis, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

Acknowledgement

This research was supported in part by the Ministry of Science and Technology, R.O.C. under grant MOST 107–2221-E-007–072-MY3 and MOST 110–2221-E-007–107-MY3.

References

[1] Blanchard M, Jacquillat A, Jaillet P. Probabilistic bounds on the *k*-traveling salesman problem and the traveling repairman problem. Math Oper Res 2023. https://doi.org/10.1287/moor.2021.0286.
[2] Cancela H, Khadiri ME, Petingi L. Polynomial-time topological reductions that preserve the diameter constrained reliability of a communication network. IEEE Trans Reliab 2011;60(4):845–51.
[3] Cancela H, Khadiri ME, Rubino G, Tuffin B. Balanced and approximate zero-variance recursive estimators for the network reliability problem. ACM Trans Model Comput Simul 2014;25(1):1–19.
[4] Cancela H, Khadiri ME, Rubino G. A new simulation method based on the RVR principle for the rare event network reliability problem. Ann Oper Res 2012;196 (1). https://doi.org/10.1007/s10479-011-1017-x.
[5] Maio FD, Coscia TM, Zio E. Data adequacy by an extended analytic hierarchy process for inverse uncertainty quantification in nuclear safety analysis. Nucl Eng Des 2024;419:112971.

[6] Su YZ, Yeh WC. The protection and recovery strategy development of dynamic resilience analysis and cost consideration in the infrastructure network. J Comput Des Eng 2022;9(1):168–86.
[7] Mahéo A, Belieres S, Adulyasak Y, Cordeau JF. Unified branch-and-benders-cut for two-stage stochastic mixed-integer programs. Comput Oper Res 2024;164:106526.
[8] Chen Z, Xiong P. RSOME in python: an open-source package for robust stochastic optimization made easy. Informs J Comput 2023;35(4). https://doi.org/10.1287/ijoc.2023.1291.
[9] Liu T, Ramirez-Marquez JE, Jagupilla SC, Prigiobbe V. Combining a statistical model with machine learning to predict groundwater flooding (or infiltration) into sewer networks. J Hydrol 2021;603:126916 (Amst).
[10] Kim YS, Pham H, Chang IH. Deep-learning software reliability model using SRGM as activation function. Appl Sci 2023;13(19):10836.
[11] Zheng J, Zhong J, Chen M, He K. A reinforced hybrid genetic algorithm for the traveling salesman problem. Comput Oper Res 2023;157:106249.
[12] Archetti C, Feillet D, Mor A, Speranza MG. Dynamic traveling salesman problem with stochastic release dates. Eur J Oper Res 2020;280(3):832–44.
[13] Kumar R, Memoria M. A review of memetic algorithm and its application in traveling salesman problem. Int J Emerg Technol 2020;11(2):1110–5.
[14] Wang Y, Ran S, Wang GG. Role-oriented binary grey wolf optimizer using foraging-following and Lévy flight for feature selection. Appl Math Model 2024;126:310–26.
[15] Rokbani N, Kumar R, Abraham A, Alimi AM, Long HV, Priyadarshini I, Son LH. Bi-heuristic ant colony optimization-based approaches for traveling salesman problem. Methodol Appl 2021;25:3775–94.
[16] Wu J, Duan Q. An algorithm for solving travelling salesman problem based on improved particle swarm optimisation and dynamic step Hopfield network. Int J Veh Des 2023;91:208–31.
[17] Habib A, Akram M. Optimizing traveling salesman problem using tabu search metaheuristic algorithm with Pythagorean fuzzy uncertainty. Granul Comput 2024;9(16). https://doi.org/10.1007/s41066-023-00424-8.
[18] Jiang Y, Tsai P, Hao Z, Cao L. Automatic multilevel thresholding for image segmentation using stratified sampling and Tabu Search. Soft Comput 2015;19:2605–17.
[19] Wang H, Li YF. Bioinspired membrane learnable spiking neural network for autonomous vehicle sensors fault diagnosis under open environments. Reliab Eng Syst Saf 2023;233:109102.
[20] Gupta M., Konar D., Bhattacharyya S., Biswas S. Computer Vision and Machine Intelligence in Medical Image Analysis International Symposium. International Symposium ISCMM 2019, 1.
[21] Bhattacharyya S, Konar D, Platos J, Kar C, Sharma K. Hybrid machine intelligence for medical image analysis. Springer; 2020, 172536.
[22] Wang YJ, Wang GG, Tian FM, Gong DW, Pedrycz W. Solving energy-efficient fuzzy hybrid flow-shop scheduling problem at a variable machine speed using an extended NSGA-II. Eng Appl Artif Intell 2023;121:105977.
[23] Liu X., Zhu W., Zhu Z., Luo L. A diverse environment coal gangue image segmentation model combining improved U-Net and semi-supervised automatic annotation. In book: cognitive systems and information processing 2023: 167–79.
[24] Li X, Li B, Tian L, Zhu W. Classification of benign and malignant pulmonary nodules based on radiomics and random forests algorithm. Huanan Ligong Daxue Xuebao. J South China Univ Technol Nat Sci 2018;46(8):72–80.
[25] Wang H, Li YF, Ren J. Machine learning for fault diagnosis of high-speed train traction systems: a review. Front Eng Manag 2023:1–17.
[26] Lai PJ, Wu HC. Using heuristic algorithms to solve the scheduling problems with job-dependent and machine-dependent learning effects. J Intell Manuf 2015;26(4):691–701.
[27] Perera J, Liu SH, Mernik M, Črepinšek M, Ravber M. A graph pointer network-based multi-objective deep reinforcement learning algorithm for solving the traveling salesman problem. Mathematics 2023;11(2):437.
[28] Hossain Z, Yeap W. How Albot1 computes its perceptual map: a theory of spatial cognition. Procedia Soc Behav Sci 2013;97:553–60.
[29] Pop PC, Cosma O, Sabo C, Sitar CP. A comprehensive survey on the generalized traveling salesman problem. Eur J Oper Res 2024;314(3):819–35.
[30] Carr R, Ravi R, Simonetti N. A new integer programming formulation of the graphical traveling salesman problem. Math Program 2023;197:877–902.
[31] Goodarzi M, Azimi M, Mohades A, Forghani-Elahabad M. Effects of different geometric patterns on free form gridshell structures. Int J Electr Comput Eng 2023;13(2):1698.
[32] Yeh WC. Novel self-adaptive monte carlo simulation based on binary-addition-tree algorithm for binary-state network reliability approximation. Reliab Eng Syst Saf 2022;228:108796.
[33] Dong X, Lin Q, Shen F, Guo Q, Li Q. A novel hybrid simulated annealing algorithm for colored bottleneck traveling salesman problem. Swarm Evol Comput 2023;83:101406.
[34] Yeh WC. Novel binary-addition tree algorithm (BAT) for binary-state network reliability problem. Reliab Eng Syst Saf 2021;208:107448.
[35] Zhang T, Sobhani B. Multi-criteria assessment and optimization of a biomass-based cascade heat integration toward a novel multigeneration process: application of a MOPSO method. Appl Therm Eng 2024;240:122254.
[36] Yeh WC, Zhu W, Yin Y, Huang CL. Cloud computing considering both energy and time solved by two-objective simplified swarm optimization. Appl Sci 2023;13:2077.
[37] Li X, Li X, Wang K, Yang S. A strength pareto evolutionary algorithm based on adaptive reference points for solving irregular fronts. Inf Sci 2023;626:658–93 (Ny).
[38] Pesode P, Barve S, Wankhede SV, Jadhav DR, Pawar SK. Titanium alloy selection for biomedical application using weighted sum model methodology. Materialstoday 2023;72(3):724–8.
[39] Lin P, Zhang L, Tiong RLK. Multi-objective robust optimization for enhanced safety in large-diameter tunnel construction with interactive and explainable AI. Reliab Eng Syst Saf 2023;234:109172.
[40] Huang PC, Tong LI, Chang WW, Yeh WC. A two-phase algorithm for product part change utilizing AHP and PSO. Expert Syst Appl 2011;38(7):8458–65.
[41] Sampath B, Sasikumar C, Myilsamy S. Application of TOPSIS optimization technique in the micro-machining process. Trends, paradigms, and advances in mechatronics engineering. IGI Global; 2023. p. 26.
[42] Yeh WC. An interactive augmented max-min MCS–RSM method for the multi-objective network reliability problem. Int J Syst Sci 2007;38(2):87–99.
[43] Khakzad N. A goal programming approach to multi-objective optimization of firefighting strategies in the event of domino effects. Reliab Eng Syst Saf 2023;239:109523.
[44] Yeh WC. A quick BAT for evaluating the reliability of binary-state networks. Reliab Eng Syst Saf 2021;216:107917.
[45] Yeh WC, Zhu W, Huang CL, Hsu TY, Liu Z, Tan SY. A new BAT and pagerank algorithm for propagation probability in social networks. Appl Sci 2022;12. https://doi.org/10.3390/app12146858.
[46] Yeh WC. Novel recursive inclusion-exclusion technology based on BAT and MPs for heterogeneous-arc binary-state network reliability problems. Reliab Eng Syst Saf 2023:231. https://doi.org/10.1016/j.ress.2022.108994.
[47] Yeh WC. A new hybrid inequality BAT for comprehensive All-level D-MP identification using minimal paths in multistate flow network reliability analysis. Reliab Eng Syst Saf 2024;244:109876.
[48] Yeh WC, Tan SY, Zhu W, Huang CL, Yang G. Novel binary addition tree algorithm (BAT) for calculating the direct lower-bound of the highly reliable binary-state network reliability. Reliab Eng Syst Saf 2022;223:108509.
[49] Yeh WC, Kuo CC. Predicting and modeling wildfire propagation areas with BAT and maximum-state PageRank. Appl Sci 2020;10(23):8349.
[50] Yeh WC, Lin E, Huang CL. Predicting spread probability of learning-effect computer virus. Complexity 2021;2021:6672630.
[51] Yeh WC. The k-out-of-n acyclic multistate-node networks reliability evaluation using the universal generating function method. Reliab Eng Syst Saf 2006;91(7):800–8.
[52] Yeh WC. A simple algorithm for evaluating the k-out-of-n network reliability. Reliab Eng Syst Saf 2004;83(1):93–101.
[53] Forghani-elahabad M, Francesquini E. Usage of task and data parallelism for finding the lower boundary vectors in a stochastic-flow network. Reliab Eng Syst Saf 2023;38:109417.
[54] Hao Z, Yeh WC, Liu Z, Forghani-elahabad M. General multi-state rework network and reliability algorithm. Reliab Eng Syst Saf 2020;202(203):107048.
[55] Forghani-elahabad M, Yeh WC. An improved algorithm for reliability evaluation of flow networks. Reliab Eng Syst Saf 2022;221:108371.
[56] Forghani-elahabad M, Kagan N. Reliability evaluation of a stochastic-flow network in terms of minimal paths with budget constraint. IISE Trans 2019;51(5):547–58.