Pamantasan ng Lungsod ng Maynila

ENHANCEMENT OF TABU SEARCH ALGORITHM FOR OPTIMIZED RESCUE ROUTING OPERATION

A Thesis Presented to the
Faculty of Computer Science Department
College of Information System and Technology Management
Pamantasan ng Lungsod ng Maynila

In Partial Fulfillment of the Requirements for the Degree
Bachelor of Science in Computer Science

By
Cando, Jhaime Jose O.
Manaois, Shelley Pe L.
Verano, Angelo P.

Mark Christopher R. Blanco
Thesis Adviser

May 2025

APPROVAL SHEET

The thesis hereto titled

ENHANCEMENT OF TABU SEARCH ALGORITHM FOR OPTIMIZED RESCUE ROUTING OPERATION

prepared and submitted by Jhaime Jose O. Cando, Shelley Pe L. Manaois and Angelo P. Verano in partial fulfilment of the requirements for the degree of Bachelor of Science in Computer Science has been examined and is recommended for acceptance and approval for **Oral Examination**.

____
Mark Christopher R. Blanco
Adviser

PANEL OF EXAMINERS

Approved by the Committee on Oral Examination
with a grade of __________ on ________.

____
Vivien A. Agustin
Thesis Coordinator

____
Raymund M. Dioses
Panel Member

____
Mark Anthony S. Mercado
Panel Member

Accepted and approved in partial fulfilment of the requirements for the degree of Bachelor of Science in Computer Science.

____
Raymund M. Dioses
CS Chairperson

____
Dr. Khatalyn E. Mata
Dean
College of Information Systems and Technology Management

ABSTRACT

The Tabu Search algorithm, while effective for optimization problems, faces challenges with getting stuck in suboptimal solutions, slow processing for large-scale issues, and inconsistent performance across different problem sizes. The TS algorithm specifically struggles in three critical areas: it often gets trapped in suboptimal solutions and cannot find the best possible solution, it suffers from high computational complexity leading to excessive runtime, and it encounters scalability issues that produce unexpected results when applied to problems of varying sizes. To address these limitations, researchers established three primary objectives: to enhance the balance of exploration and exploitation by incorporating a modified perturbation technique to diversify the search space, to optimize the high-complexity operations by swapping selected Points of Interest (POIs) to speed up the generation of neighborhood solutions, and to improve scalability by utilizing a modified dynamic tabu tenure mechanism that adjusts parameters based on problem size. The methodology implemented three innovative approaches: first, a wave-resonance based perturbation technique which mimics wave interference patterns observed in transmission frequency analysis to better explore the solution space; second, a focal point sampling method for neighborhood generation that strategically examines only the most promising route segments rather than exhaustively checking every possibility; and third, a Wave-Inspired Dynamic Tenure mechanism (WIDT) that dynamically adjusts memory length using principles derived from wave properties. Testing demonstrated significant improvements over previous approaches: solution diversity increased substantially (scoring 0.21 versus 0.03), processing time for 160-location problems became eight times faster (3.33 seconds versus 28.88 seconds), and performance remained consistent across all tested scenarios regardless of size.

ACKNOWLEDGEMENTS

The success of this study is due to the tremendous encouragement and assistance from many people and institutions. We want to express our sincere gratitude to everyone who inspired and helped us along the way.

First and foremost, we thank our Lord for providing us with insight and direction throughout this research project. We are incredibly grateful to our parents, whose steadfast love and sacrifices fueled our academic and personal development. We also recognize the support and encouragement we received from our friends and family, whose words enabled us to overcome challenges.

We would especially like to thank Mr. Mark Christopher R. Blanco, our thesis adviser, and Ms. Khatalyn E. Mata for their excellent advice and criticism, which greatly improved our research. The panelists, Professors Raymund M. Dioses and Vivien A. Agustin, are also appreciated for their valuable comments throughout the defense, which improved the caliber of our research.

To Alma Mater, Pamantasan ng Lungsod ng Maynila, PLM College of Information System and Technology Management (PLM CISTM), in accordance with the requirements of the Bachelor of Science in Computer Science.

- The Researchers

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter One

INTRODUCTION

1.1 Background of the Study

Tabu Search is a meta-heuristic that directs a heuristic search procedure to explore solutions beyond local optimality making it a widely favored method, for various issues The key idea, behind Tabu search is the importance of including adaptive memory and responsive exploration in any intelligent search approach Adaptive memory facilitates efficient and effective exploration of solution space enhancing the overall search effectiveness Tabu search methods are characterized by their use of memory based tactics that aim to blend memory forms with strategies, for optimization—an endeavor rooted in the idea of integrating various principles effectively (Andhale, 2024). By incorporating a memory feature into tabu search (TS) it becomes possible to employ techniques that can navigate the solution space with both efficiency and economy. Unlike memoryless approaches that depend largely upon random processes involving sampling methods, for decision making purposes request choices made in TS are influenced by information gleaned throughout the search process (GeeksforGeeks, 2024). Tabu search uses local or neighborhood iterative procedures to move from solution x to a solution x' (in the vicinity of x) until the stopping conditions are met

Tabu Search is one of the widely used metaheuristics to solve complex optimization problems. However, it faces several issues that limit its effectiveness. First and foremost, it is often trapped by local optima due to insufficient exploration and hence generates suboptimal solutions. Also, in many iterations, it does nothing but waste the computational resources and become inefficient. Further, the static nature of the tabu tenure creates scalability problems; therefore, the algorithm is less favorable for larger and dynamic problem sets (Wang et al., 2016). Such problems demand improvement in the algorithm for performance and adaptability.

To overcome such challenges, this research aims to improve the Tabu Search algorithm. The objectives are as follows: improvement of the exploration ability to avoid stagnation in local optima, adaptive mechanism to reduce unnecessary iterations, and a dynamic tabu tenure to ensure scalability. By achieving these improvements, the study intends to develop an efficient and reliable optimization tool for real-world applications, especially in critical scenarios, such as rescue routing operations.

This paper attempts to improve the Tabu Search algorithm in terms of optimizing rescue routing operations in the Baseco Compound, Manila, Philippines. Rescue efficiency can be improved through developing a routing solution that will favor only those houses not evacuated. The importance of rescue operations means that optimal route planning should always ensure that response times and resource distribution are appropriate. Although this study has an operational scenario limited to a single study case and does not provide vehicle resources usage in rescue operations

1.2 Statement of the Problem

The Tabu Search (TS) algorithm has been widely applied to solve complex optimization problems, but it suffers from critical limitations that hinder its effectiveness, particularly in large-scale and dynamic scenarios.

Input: $soln_{init}$

Output: $soln_{best}$

$tabu_{list} \leftarrow$ empty list
$soln_{curr} \leftarrow soln_{init}$
$soln_{best} \leftarrow soln_{init}$
For all $iter_{ctr} \leftarrow 1$ **to** $iter_{max}$ **do**
    $nbhd \leftarrow$ empty list
    For all $p1$ **in** $soln_{curr}$ **do** ➤ SOP 2
        For all $p2$ **in** $soln_{curr}$ **do**
            $soln_{mod} \leftarrow soln_{curr}$
            $soln_{mod} \leftarrow soln_{curr}[p1 \leftrightarrow p2]$ ➤ SOP 1
            $nbhd \leftarrow nbhd \cup soln_{mod}$
        End for
    End for
    $nbhr_{best} \leftarrow$ BestAdmissibleSoln($nbhd$)
    If Val($nbhr_{best}$) < Val($soln_{best}$) **then**
        $soln_{best} \leftarrow nbhr_{best}$
    End if
    $tabu_{list} \leftarrow tabu_{list} \cup nbhr$
    If $|tabu_{list}| > tabu_{tenure}$ **then** ➤ SOP 3
        RemoveFirst($tabu_{list}$)
    End if

Figure 1.1 Algorithm of Current Tabu Search

As outlined in Algorithm 1, the TS algorithm's core operations involve generating a neighborhood of solutions, selecting the best admissible solution, and updating the tabu list. However, this approach often leads to:

1. **The TS algorithm often gets stuck in suboptimal solutions, unable to find the best possible solution.**

This is largely due to its imbalance between exploration and exploitation, as highlighted in study by Zhao et al. (2023). The TS algorithm tends to prioritize exploitation, leading to premature convergence.

Table 1.1. Diversity Score of 3 TS algorithms (40 POI)

Variant	**Diversity Score (CV)**
Zhao et al. (2023)	0.02 - 0.03
Wang et al. (2016)	0.07
Khalid et al. (2024)	0.03 - 0.06

Note: Variant: Variant of TS algorithm: Diversity Score - Uses Coefficient Variation

The data presented in Table 1.1 shows that the Zhao et al. (2023) Tabu Search algorithm variant has a low diversity (0.02 - 0.03), indicating it doesn't effectively explore different possible solutions. This limitation is specifically acknowledged in their own study, confirming the algorithm's restricted ability to find diverse options (Zhao et al., 2023).

The findings in Table 1.1 further demonstrate that Zhao et al. (2023) probabilistic approach produces results similar to other TS variants, which indicates a similar problem. This lack of consistent improvement over existing methods makes their approach unsuitable for real-world applications where finding high-quality, reliable solutions is essential.

2. The TS algorithm have high-complexity leading to high runtime

The TS algorithm is widely recognized for its high computational expense, as highlighted in study by Wang et al. (2016). The TS algorithm tends to have significant computational challenges, particularly as problem size increases.

Table 1.2. Wang et al. (2016) Variant Simulation Result

POI Size	**Runtime**
40	0.41s - 0.53s
80	3.41s - 3.51s
160	27.73s - 28.88s

Note: Runtime - Average duration of each run in seconds

The simulation conducted by the researchers at Table 1.2 clearly shows how the variant of the TS algorithm conducted by Wang et al (2016) takes significantly longer to run. For POI: 160, it needs about 27.73s-28.88s seconds indicating a high complexity time for larger POI.

More recent studies by Khalid et al. (2024) and Zhao et al. (2023) did manage to make things run faster, but at a cost - their solutions weren't as good. Naraharisetti and Raghuveer specifically pointed out that their variant of TS algorithm needs smarter ways to find good solutions quickly. This shows a common challenge in these algorithms: you can make them fast or you can make them accurate, but doing both at once remains difficult.

3. **The TS algorithm has a scalability issue that leads to unexpected results.**

This scalability issue is evident in numerous studies, as highlighted in study by Khalid et al. (2024). The TS algorithm has identified static parameter settings as a primary contributor to this problem.

Table 1.3. Khalid et al. (2024) Variant Simulation

POI Size	**Avg Solution**	**Diversity Score (CV)**
40	1716.49 - 1774.13	0.03 - 0.06
80	4061.92 - 4124.72	0.04 - 0.05
160	8143.39 - 8162.37	0.02

Note: Avg. Solution - Solutions in each iteration: Diversity Score - Uses Coefficient Variation

The data in Table 1.3 reveals that Khalid et al.'s (2024) version of the TS algorithm shows a low diversity score as POI increases. This suggests that there is a substantial decline in diversification, indicating serious problems with handling larger datasets.

This scaling weakness is also evident in the work of Wang et al. (2018) and Zhao et al. (2023). The consistently poor quality of solutions from these approaches makes them unsuitable for real-world use, particularly when dealing with larger, more complex problems.

1.3 Objective of the Study

To address the identified limitations of the TS algorithm, this study aims to develop innovative strategies to improve its performance. By enhancing the algorithm's exploration capabilities, reducing computational cost, and improving scalability, the researchers seek to make the TS algorithm a more effective tool for solving complex optimization problems.

1.3.1 General Objective

The primary goal of this study is to improve the performance of the Tabu Search algorithm, especially for large and complex problems.

1.3.2 Specific Objectives

Specifically, the study aims to perform the following:

1. To enhance the balance of exploration and exploitation by incorporating modified perturbation technique to diversify the search space.
2. To optimize the high-complexity by swapping selected POIs to speed up the generation of neighborhood.
3. To improve scalability by utilizing modified dynamic tabu tenure mechanism to adjust the parameter based on problem size

1.4 Significance of the Study

This section discusses the importance of this research. It highlights the potential benefits and implications of the study's findings.

Emergency Response Teams: They can benefit from the system by improving response times, allocating resources efficiently, and coordinating efforts effectively.

Affected Communities: They can directly benefit from faster relief efforts and minimized suffering due to optimized rescue operations.

Industry Practitioners: Professionals in industries requiring optimization solutions, such as logistics, supply chain management, and rescue operations, can utilize the enhanced Tabu Search algorithm for practical applications.

Academic Institution: The institution can benefit from this study as it can be used as a learning material. The study will serve as valuable learning material for students and educators interested in algorithm design, optimization techniques, and their real-world applications.

Future Researchers: This study will benefit future researchers who gained interest to use the enhancement of tabu search algorithm as part of their study. This study will also benefit future researchers as reference for their research.

1.5 Scope and Limitations

This section explains the scope and limitations of the study. It identifies the specific focus areas and acknowledges the factors that may restrict the study.

The study will focus on enhancing the TS algorithm by addressing issues identified in recent studies. In particular, the research aims to provide information about rescue route operation within Baseco Compound, Manila, Philippines. The primary data source is the data set provided by the Manila Geographic Information, and any locations not included in the dataset may be manually added using coordinates. Furthermore, the study is limited to a single operation.

Certain limitations are unavoidable in the study such as the rescuer's resources like maximum capacity and technical issues like the absence of Wi-Fi connectivity will not be considered. The system in this study utilizes Google's Distance Matrix API and Directions API with a free account, which imposes limitations such as maximum number of entries (e.g., 25 locations in a 12 by 12 format). Additionally, the number of data points provided by the Manila Geographic Information is determined by them.

1.6 Definition of Terms

The following terms are defined to provide clarity and establish a common understanding of the concepts used throughout this study. Familiarity with these terms will facilitate a better understanding of the methodology, implementation, and results discussed in subsequent sections.

Adaptive Mechanism: A system that automatically adjusts settings (like memory length) to improve efficiency as the problem evolves.

Adaptive Memory: The ability of Tabu Search to remember past decisions and their outcomes, helping it choose better future steps.

Adaptive Stopping Conditions: Rules that tell Tabu Search when to stop running, based on real-time progress or changes.

Algorithm: Step-by-step instructions to solve a problem or complete a task, often used in computer programs.

Combinatorial Issues: Problems where you must find the best combination or arrangement among many possibilities.

Dynamic Environment: A situation that changes over time, requiring strategies to adapt as new challenges arise.

Exploitation: Improving existing solutions by focusing on and refining their strongest parts.

Exploration: Searching for new solutions in different areas to avoid getting stuck in familiar patterns.

Focal Point Sampling: Observing a single animal for a set time and noting all its actions with timestamps (used in behavioral studies).

Local Optima: A solution that works best in a small area of possibilities but might not be the best overall.

Oscillate: To swing or move back and forth repeatedly at a steady pace.

Perturbation: Introducing small, deliberate changes to a solution to test new ideas or avoid repetition.

Points of Interest (POI): Important spots in a problem's solution space that could lead to better answers.

Quantum: The smallest possible unit of a physical property, like energy or charge.

Quantum Entanglement: A connection between particles where their states are linked, even if they're far apart.

Resonance: A strong vibration in an object when it matches the rhythm of an external force (e.g., a guitar string vibrating at a specific pitch).

Routing: Planning the most efficient path to deliver resources (e.g., emergency teams) to where they're needed.

Solution: The answer or result produced by solving a problem with an algorithm.

Tabu Search: A problem-solving method that uses memory to avoid repeating past mistakes and explore new options.

Wave: A pattern of movement that travels from one place to another in an organized way

Chapter Two

REVIEW OF RELATED LITERATURE

This chapter delves into Related Literature of the study by reviewing relevant recent studies and literatures. Gaining understanding of problems that are relevant to and solutions used made easier by the information in this chapter.

2.1 Related Literature

This section outlines the methodology of the study by reviewing relevant literature. It explores key concepts, theories, and principles that underpin the research problem.

Tabu Search | Artificial intelligence

The article by Andhale (2024) discusses the application of Tabu Search, a metaheuristic algorithm, in solving optimization problems, particularly emphasizing its memory mechanism. The key contribution of Tabu Search is its ability to avoid revisiting poor solutions through the use of a "tabu list," which records recent moves. Despite its strengths, the study acknowledges a significant limitation in the practical implementation of Tabu Search. Tuning multiple algorithm parameters, such as tabu tenure and the length of the tabu list, remains a challenge. As Andhale (2024) states, "It is challenging to tune the multiple parameters that make up an algorithm." This issue impacts the algorithm's efficiency and performance across diverse problem instances, and the study does not offer a definitive solution to this limitation. Further research is needed to establish systematic methods for parameter tuning to enhance the algorithm's applicability and reliability in varying contexts.

Tabu Search Implementation on Traveling Salesman Problem and Its Variations: A Literature Survey

The study by Basu (2012) investigates the application of Tabu Search (TS) for solving the Traveling Salesman Problem (TSP) and its variations, such as the Vehicle Routing Problem (VRP). A critical issue identified in the study is the challenge of applying Tabu Search effectively to large problem sizes, particularly asymmetric instances of the TSP. Basu (2012) states, "The issue of deciding tabu tenures has not received adequate attention in the literature. In aggregate, more authors have preferred fixed tabu tenures over random tabu tenures. This preference seems to have increased in recent years. The problem comes while implementing tabu search on large problem sizes because commonly used fixed tenure values tend to fail since the neighborhood size is large" (p. 169). The study emphasizes that while Tabu Search demonstrates significant utility, the inability to tailor tabu tenures for varying problem sizes remains an unresolved challenge. Further research is necessary to explore functional dependencies beyond simple linear or logarithmic models.

Trade-offs between Exploration and Exploitation in Local Search Algorithms

In the context of optimization algorithms, the balance between exploration and exploitation is crucial for achieving optimal solutions. Local search algorithms, such as Simulated Annealing and Tabu Search, illustrate this trade-off, where exploration helps to avoid local optima by diversifying the search, while exploitation focuses on refining existing solutions. Effective balancing ensures that algorithms do not become trapped in suboptimal solutions or waste computational resources. Metaheuristic strategies, including genetic algorithms and particle swarm optimization, employ adaptive techniques to manage this balance dynamically based on search feedback. Furthermore, the CLEGE algorithm exemplifies a structured approach that integrates local exploitation with global exploration, enhancing search efficiency. However, achieving the right balance can be challenging due to factors such as problem complexity and dimensionality, which necessitate an adaptable strategy tailored to specific optimization scenarios (GeeksforGeeks, 2024).

Time-Windowed Vehicle Routing Problem: Tabu Search Algorithm Approach

Demir et al. (2022) postulate that coupling TS with machine learning may provide better dynamic parameter adaptation and real-time decision-making. Second, more research into hybrid TS models, e.g., integrating with genetic algorithms or variable neighborhood search, is also needed to further enhance computational performance in large-scale routing problems. These developments are essential to fulfilling the growing requirement for adaptive and scalable logistics services.

Tabu Search Algorithm: Optimizing the search runtime

Khalid et al. (2024) proposed a new data structure, Drop-set, to enhance tabu list management, lowering the computational burden in Tabu Search (TS) considerably. In spite of enhanced computational performance, TS is still not scalable for huge rescue operations. Static tabu tenures induce unnecessary iterations and wastefulness, necessitating the need for real-time adaptability. Khalid et al. (2024) propose that reinforcement learning is to be blended with TS in order to provide improved decision-making during emergency interventions. Research is to be concentrated on hybrid systems combining TS and AI-based adaptive strategies for more effective real-time rescue routing.

Large-scale timetabling problems with adaptive tabu search

Tabu Search (TS) has been investigated as a strong optimization method for resource planning in dynamic environments, such as rescue missions. Awad et al. (2022) proved the effectiveness of TS in solving large-scale timetabling issues through cycle avoidance and enhanced local optimization. Though TS improves the quality of solutions during rescue scenarios, its efficiency for computation degrades with larger instance sizes of problems. To manage exploration and exploitation, Awad et al. (2022) propose blending adaptive heuristics with dynamic tabu tenure mechanisms. Further research is to be oriented toward the integration of real-time environmental parameters like traffic and the magnitude of the disaster to facilitate enhanced applicability of TS for larger-scale real-time rescue routing tasks. Enhanced memory structures and scalability are important issues for effective application.

Trade-offs between Exploration and Exploitation in Local Search Algorithms

Contemporary research in routing optimization increasingly focuses on developing intelligent algorithms capable of balancing computational efficiency with solution quality (GeeksforGeeks, 2024). By integrating adaptive parameter tuning and memory-based search mechanisms, modern local search techniques aim to create more robust routing strategies that can quickly adapt to changing network conditions while consistently identifying near-optimal solution paths.

Focal Animal Sampling

Focal animal sampling is a well-established observational technique in behavioral research, wherein a single subject is intensively observed to document its behavior in detail over a specified period. The methodological principles of focal animal sampling also hold significant promise when applied to the field of optimization. By concentrating on the most critical and informative components—as illustrated in the study—this targeted approach could be adapted to prioritize key elements that substantially influence overall outcomes in optimization problems. Such a strategy suggests that focusing analytical efforts on these pivotal areas may lead to more efficient exploration of complex solution spaces and ultimately facilitate faster convergence and enhanced performance (Bosholn & Anciães, 2022).

Exploiting Learned Policies in Focal Search

The article "Exploiting Learned Policies in Focal Search" describes the use of learned policies to improve the performance of Focal Search (FS), which is a heuristic search algorithm that chooses candidate solutions from a finite set of promising solutions. With the incorporation of policies learned by machine learning models, the algorithm directs the search dynamically in directions with greater potential for optimal solutions. The method efficiently minimizes computational complexity and enhances search effectiveness by concentrating on promising solution directions, which is highly beneficial for solving large-scale combinatorial optimization problems. Given that Focal Search is similar to Tabu Search in iterative improvement of solutions and exploration of solutions, this method can be employed to eliminate the high runtime and complexity problems of TS, particularly in routing problems (Araneda et al., 2021).

Applying learned policies to Tabu Search can include adaptive decision-making that adjusts the exploration-exploitation balance dynamically based on real-time feedback. This adaptability can counteract local optima trapping and suboptimal search due to fixed tabu list configuration. By learning from previous iterations of search, the algorithm can identify patterns and concentrate on solution directions with higher potential, thereby maximizing computational efficiency and solution quality. This enhancement fits your objective of improving Tabu Search for routing problems by maximizing runtime and solution optimality, and the use of learned policies is therefore an interesting avenue for future research (Araneda et al., 2021).

Accurate numerical analysis of resonances in random waveguides: Effects of the waveguide length

The study (Katayama et al., 2024) presents a detailed numerical investigation of resonance phenomena in random waveguides. Drawing inspiration from the physical resonance mechanisms explored in the study, the idea of integrating wave-resonance concepts into optimization algorithms holds significant potential. In optimization, particularly in metaheuristic methods like tabu search, introducing adaptive perturbations that mimic the periodic and localized nature of wave resonances can help maintain a balance between exploration and exploitation. This resonance-inspired approach could facilitate escaping local minima by dynamically adjusting perturbation intensities based on progress and stagnation metrics, mirroring how resonances persist regardless of system size. The study's insights into robust, geometry-independent resonance behavior suggest promising avenues for developing more efficient and resilient optimization strategies.

Quantum Mechanics

Tabu Search is a method used to solve routing problems by avoiding repeated solutions through a fixed "forbidden period." However, this rigidity can limit its efficiency. Previous studies have explored adjusting this period manually, but these methods lack adaptability. To address this, the proposed thesis introduces a dynamic approach inspired by quantum mechanics, specifically quantum waves, which govern probabilistic changes in particle behavior (Ismael, 2025). By mimicking how quantum systems adapt based on probabilities, the new method adjusts the forbidden period in real-time. For example, during active exploration of new routes, the period shortens to encourage flexibility, while it lengthens to prevent repetition when the search stagnates.

This innovation aims to balance exploration and exploitation more effectively than traditional methods. The quantum-inspired adjustment aligns with the natural adaptability seen in physical systems, offering a smarter way to optimize routing operations. By dynamically changing the forbidden period, the enhanced Tabu Search could improve route planning in logistics, leading to faster, cost-effective solutions. The study's focus on simplicity and real-world application addresses a key gap in existing optimization techniques, providing a practical tool for complex routing challenges.

2.2 Related Studies

This section outlines a comprehensive review of empirical studies that are directly related to the research topic. It examines previous research findings, methodologies, and conclusions to identify gaps in knowledge.

Novel hybrid algorithm for Team Orienteering Problem with Time Windows for rescue applications

The study by Saeedvand et al. (2020) effectively integrates Multi-Objective Evolutionary Algorithms (MOEAs) with Q-learning and artificial neural networks (ANNs) to address the Team Orienteering Problem with Time Windows in Rescue (TOPTWR). Notable limitation is the lack of real-time task prioritization for critical tasks. Although the study emphasizes the varying importance of tasks in a rescue environment, such as prioritizing saving lives over other objectives, the proposed method does not fully address real-time adaptability in these scenarios. The authors mention, “In a disaster rescue environment, different tasks may have different importance. For instance, saving two people in the same room is more important than saving a person in one room or cutting electricity can be more important than opening a window.” Furthermore, the algorithm’s reliance on computationally intensive processes, such as MOEAs, Q-learning, and ANNs, increases the resource demands for training and optimization. This issue limits its feasibility for real-time applications in resource-constrained environments. The authors highlight this challenge, stating that “using a neural network as a function approximation, suitable solutions can be achieved. … However, based on the nature of rescue applications, dynamicity in the environment is inevitable.” These unresolved issues suggest opportunities for further enhancements to improve the algorithm’s real-time applicability and scalability in dynamic and large-scale rescue operations.

ILSGVCP: An improved local search algorithm for generalized vertex cover problem

The generalized vertex cover problem is an important and difficult extension of the minimum vertex cover problem, and was classified to be NP hard, meaning that it is very computationally intensive and it can't be solved efficiently in reasonable times when the size of the graph grows or the number of constraints increases. Unlike the minimum vertex cover that focuses on covering all the edges using the minimum number of vertices, the generalized application of this problem introduces other constraints, which widen its applicability but at the same time magnify its complexity. These create quite challenging problems for areas such as network design, resource allocation, bioinformatics, and data clustering because of the need for precise and efficient solutions in solving them (Tai et al., 2023). A notable challenge in the study is adjusting the perturbation mechanism’s strength to suit different problem sizes. As mentioned, "the size of perturbation strength will greatly influence perturbation effect. The perturbation strength is too large will cause the restart. The perturbation strength is too small will not be able to jump out of local optima, or it is easy to fall into cycling," indicating that improper tuning can hinder solution quality. This difficulty in optimizing the perturbation mechanism limits LSTP’s adaptability to various GVCP instances, pointing to a need for further refinement in parameter tuning for more consistent performance

A Tabu Search algorithm for the Probabilistic Orienteering Problem

The study by Chou et al. (2020) presents an enhanced Tabu Search (TS) algorithm, incorporating Monte Carlo sampling to tackle the Probabilistic Orienteering Problem (POP). While the enhanced algorithm demonstrates effective solution quality, the researchers encountered a key challenge: the complexity of the objective function increases with the number of required samples in Monte Carlo approximation, which, while improving solution accuracy, also demands more computational time. They note, "the approximation of the objective function value u(τ) is more accurate when more scenarios are used, but this requires more computational time," highlighting a trade-off between accuracy and speed (Chou et al., 2020)

Parallelization of dial-a-ride using tabu search

The study's findings emphasize the inherent trade-off in parallel Tabu Search between computational efficiency and solution quality (Zhao et al., 2023). While parallel processing aims to speed up the search, the scarcity of impactful moves limits its effectiveness. This aligns with broader challenges in optimization algorithms, where expanding the search space does not always translate to improved results. The authors' work underscores the need for smarter strategies to identify high-quality moves efficiently, rather than relying solely on increased computational power.

An Improved Tabu Search Algorithm for Solving Heterogeneous Fixed Fleet Open Vehicle Routing Problem with Time Windows

Ahmed and Yousefikhoshbakht (2022) developed an enhanced Tabu Search algorithm that combines a mixed integer linear programming model with advanced local search techniques to address the heterogeneous fixed fleet open vehicle routing problem with time windows. While their approach outperforms traditional methods and exact algorithms in delivering high-quality near-optimal solutions, the study identifies a persistent challenge: certain complex problem instances still require excessively long computational times. This unresolved issue highlights the need for further optimization to balance solution accuracy with efficiency in highly constrained routing scenarios.

Genetic tabu search for the fuzzy flexible job shop problem

Palacios et al. (2014) introduced a novel approach for tackling scheduling challenges involving uncertain task durations and adaptable machinery by merging genetic algorithms with Tabu Search. While the algorithm achieved superior results on established test cases—even setting new performance benchmarks—it faces limitations due to its resource-intensive search process. The complexity of evaluating numerous potential solutions hinders its application to larger or more intricate real-world scenarios, a critical issue the study leaves unaddressed.

An Adaptive Tabu Search Algorithm for Solving the Two-Dimensional Loading Constrained Vehicle Routing Problem with Stochastic Customers

Zhang et al. (2023) introduce an adaptive tabu search algorithm that integrates a multi-order bottom-fill-skyline packing heuristic with Monte Carlo simulation to tackle a two-dimensional loading-constrained vehicle routing problem under customer uncertainty, achieving competitive routing and packing performance across various loading configurations. However, the study leaves unresolved challenges regarding computational efficiency and scalability, as the complexity induced by integrating stochastic customer presence with detailed two-dimensional loading constraints remains a critical issue that the proposed method does not fully overcome.

Exponential extrapolation memory for tabu search

Bentsen et al. (2022) propose a novel adaptive memory mechanism for tabu search, termed exponential extrapolation memory, which tracks the frequency and recency of variable assignments in recent local optima to guide move selection and diversify the search process. However, while their computational study demonstrates performance improvements—particularly for static move evaluations—the impact of this memory mechanism on dynamic move evaluation and its overall influence on search behavior across diverse problem structures remains unresolved, indicating that further investigation is needed to fully understand and optimize its scalability and efficacy

MTS-PRO2SAT: Hybrid Mutation Tabu Search Algorithm in Optimizing Probabilistic 2 Satisfiability in Discrete Hopfield Neural Network

Chen et al. (2024) developed MTS-PRO2SAT, a combined mutation and Tabu Search method, to enhance probabilistic 2-satisfiability optimization in discrete Hopfield neural networks. By merging mutation-driven exploration with Tabu Search's avoidance of repetitive solutions, the algorithm improves global convergence and diversifies outcomes during training. While it outperforms existing metaheuristic methods in efficiency and solution accuracy, the study identifies unresolved limitations: sensitivity to parameter settings and struggles with scalability as network dimensions expand. These issues highlight the need for additional refinements to mitigate local solution traps and ensure reliability in real-world, large-scale implementations.

A Neighborhood Expansion Tabu Search Algorithm Based On Genetic Factors

Wang et al. (2016) developed the "Neighborhood Expansion Tabu Search Algorithm Based on Genetic Factors" (NETS) to address the Traveling Salesman Problem (TSP). This hybrid approach merges Tabu Search's local optimization efficiency with Genetic Algorithm techniques, such as crossover and mutation, to broaden solution exploration. While NETS demonstrated improved search capabilities, the study highlighted a critical unresolved issue: the algorithm's computational overhead increased by over 20% compared to traditional Tabu Search as node counts grew. The authors explicitly noted this scalability challenge, emphasizing the need for future optimizations to reduce time complexity and enhance practicality for large-scale, time-sensitive applications.

2.3 Synthesis

Recent research has explored many improvements to the Tabu Search (TS) algorithm to address a variety of optimization problems. The study Parallelization of Dial-a-Ride Using Tabu Search by Zhao et al. (2023) sought to optimize the Dial-a-Ride Problem (DARP) by applying parallel computing to the TS algorithm, thus reducing computational time and solution efficiency. Similarly, A Neighborhood Expansion Tabu Search Algorithm Based on Genetic Factors by Wang et al. (2016) introduced a hybrid approach that uses GA concepts, allowing for better exploration and quicker convergence for large data sets. Additionally, Tabu Search Algorithm: Optimizing the Search Runtime by Khalid et al. (2024) proposed the Drop-set data structure to optimize tabu list maintenance, resulting in drastic reduction of computational overhead and improvement in search efficiency. These studies indicate a preoccupation with enhancing TS using computational optimization, hybridization with other metaheuristics, and memory-efficient data structures.

Despite these advancements, there are still many challenges. The parallelization approach in Parallelization of Dial-a-Ride Using Tabu Search had an issue with move generation balance, where more parallel operations did not necessarily translate to improved runtime efficiency (Zhao et al., 2023). The Neighborhood Expansion Tabu Search Algorithm Based on Genetic Factors had high computational effort with the additional genetic operations, and it was less effective for large-scale problems (Wang et al., 2016). Likewise, although the Drop-set structure in Tabu Search Algorithm: Optimizing the Search Runtime highly optimized computational effort, the algorithm was still plagued with an issue of finding an actual global optimum with increasing problem sizes (Khalid et al., 2024).

2.4 Comparative Analysis

This section provides a comparative analysis of various algorithms. By examining the performance metrics of these algorithms, this section aims to identify the most suitable approach for the current research problem.

Table 2.1 Comparative Analysis of Various Algorithm

Meta-Heuristic	**Classification**	**Exploration vs. Exploitation**	**Memory Usage**	**Applicability**
Tabu Search (TS)	Memory-Based / Local Search	Strong in exploitation but weak in exploration, requiring diversification techniques.	Uses a tabu list to prevent revisiting previous solutions.	Best for scheduling, routing, and resource allocation.
Simulated Annealing (SA)	Probability-Based / Single-Solution	Good at exploration but slow convergence.	Memoryless.	Used in scheduling and optimization.
Genetic Algorithm (GA)	Population-Based / Evolutionary	Good exploration but can prematurely converge.	Memoryless.	Applied in machine learning and engineering.
Ant Colony Optimization (ACO)	Population-Based / Swarm Intelligence	Strong exploration but stagnates in large-scale problems.	Memory-based (pheromone trails).	Best for logistics and traffic routing.
Greedy Randomized Adaptive Search (GRASP)	Constructive / Iterative Local Search	Balances exploration and exploitation.	Memoryless.	Used in combinatorial optimization.
Particle Swarm Optimization (PSO)	Population-Based / Swarm Intelligence	Good exploration but prone to early stagnation.	Memoryless.	Used in function and power grid optimization.

Tabu Search (TS) is a powerful meta-heuristic optimization technique that is good at exploitation by searching the solution space in a systematic and non-repetitive fashion. It uses a memory structure known as a "tabu list" to prevent backtracking to already visited solutions and hence to deliver a more focused search that escapes local optima. It is highly effective in solving difficult problems like scheduling, routing, and resource allocation, where solution refinement is critical for optimal performance.

TS is unique among other meta-heuristics in that it is structured in the utilization of its memory, unlike SA or GA, which are memoryless. While SA is biased towards exploration and GA provides multiple solutions through evolution, both are prone to getting stuck in slow convergence or premature stagnation. ACO and PSO exploit swarm intelligence but may be tested in dealing with large problems. TS is unique in balancing exploitation and intelligent diversification and hence is very effective in dealing with combinatorial optimization problems requiring exact, high-quality solutions.

Chapter Three

METHODOLOGY

This chapter explains the plan used to meet the study's goals. It describes the step-by-step method created for the research, the system setup, and how simulations were used to test the ideas.

3.1 Research Design

This research adopts an experimental design approach to systematically evaluate the enhanced Tabu Search (TS) algorithm under controlled conditions. The methodology implements a factorial experimental design, manipulating variables such as dataset size (40, 80, and 160 points of interest) and algorithm variants to measure specific dependent variables including solution quality, processing time, and scalability. These controlled experiments were designed based on actual rescue operation requirements from the Baseco Compound incident involving approximately 40 individuals (AFPAO, 2024), allowing for realistic parameter settings while maintaining experimental rigor. To enhance external validity, the experiments incorporate real-world geographic data from the Manila Geographic Information System & Data Hub (Asakura et al., 2020), bridging the gap between laboratory testing and practical application while ensuring the statistical significance of performance improvements can be validated across multiple trials.

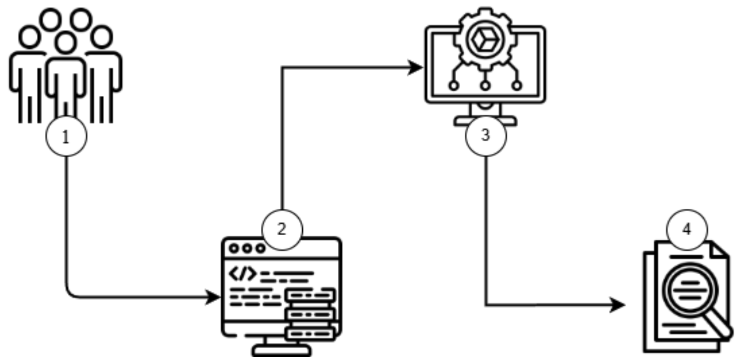

Figure 3.1 Research Design Steps

The research design employs a structured four-phase methodology to systematically enhance the Tabu Search (TS) algorithm for rescue routing optimization, as illustrated in Figure 3.1. This framework addresses three critical limitations of conventional TS implementations - premature convergence, computational inefficiency, and static parameterization - through an integrated approach combining algorithmic innovation, rigorous testing protocols, and real-world validation.

First, algorithm implementation involves creating three recent established variants of the TS algorithm and developing an enhanced version that integrates three new unique solutions. This aligns with simulation-based studies that emphasizes iterative refinement of methods to address real-word challenges (THIS institute - The Healthcare Improvement Studies Institute, 2023).

Second, simulation experiments are designed to test all algorithms under multiple runs and parameter configurations. Performance metrics–Including average solution quality, runtime, and Coefficient of Variation (Peterka, 2024)–are recorded to ensure robust comparison (Farsani, 2024).

Third, data collection captures outputs from both simulated and real-world datasets, while analysis focuses on statistical comparisons to identify strengths and limitations of the enhanced TS algorithm relative to existing variants. This approach reflects the emphasis on reproducibility and transparency in simulation research (Elendu et al., 2024).

3.2 Proposed Algorithm and System Architecture

3.2.1 Proposed Algorithm

Input: $soln_{init}$

Output: $soln_{best}$

$tabu_{list} \leftarrow$ empty list

$soln_{curr} \leftarrow soln_{init}$

$soln_{best} \leftarrow soln_{init}$

$stagnant_{ctr}, improve_{ctr} \leftarrow 0$

$tabu_{tenure} \leftarrow |soln_{init}| \times 0.1$

For all $i \leftarrow$ **to** $iter_{max}$ **do**

    If $stagnant_{ctr}$ **or** $improve_{ctr} > 10$ **then**

        $tabu_{tenure} \leftarrow$ WIDT(...) ➤ OBJ 3

    End if

    If $stagnant_{ctr} > 0$ **then**

        $soln_{curr}, stagnant_{ctr} \leftarrow$ WRP(...) ➤ OBJ 1

    End if

    $nbhd \leftarrow$ FPSN($soln_{curr}, tabu_{list}, stagnant_{ctr}$) ➤ OBJ 2

    $nbhr_{best} \leftarrow$ BestAdmissibleSoln($nbhd$)

    If Val($nbhr_{best}$) $<$ Val($soln_{best}$) **then**

        $soln_{best} \leftarrow nbhr_{best}$

        $stagnant_{ctr} \leftarrow 0$

    Else

        $stagnant_{ctr}{+}{+}$

    End if

    $soln_{curr} \leftarrow nbhr_{best}$

    $tabu_{list} \leftarrow tabu_{list} \cup nbhr_{best}$

    If $|tabu_{list}| > tabu_{tenure}$ **then**

        RemoveFirst($tabu_{list}$)

    End If

End For

Figure 3.2: Algorithm of Enhanced Tabu Search

Figure 3.2 presents the researchers' enhanced algorithm to solve the said problems, which integrates three distinct algorithms to address the problem at hand. This hybrid method employs wave-resonance perturbation to boost exploration, focal point sampling to accelerate neighborhood generation, and a wave-inspired dynamic tenure mechanism to ensure scalability. It is also shown in Figure 3.2 the solution's affected area on the left side of the schema.

3.2.2 System Architecture

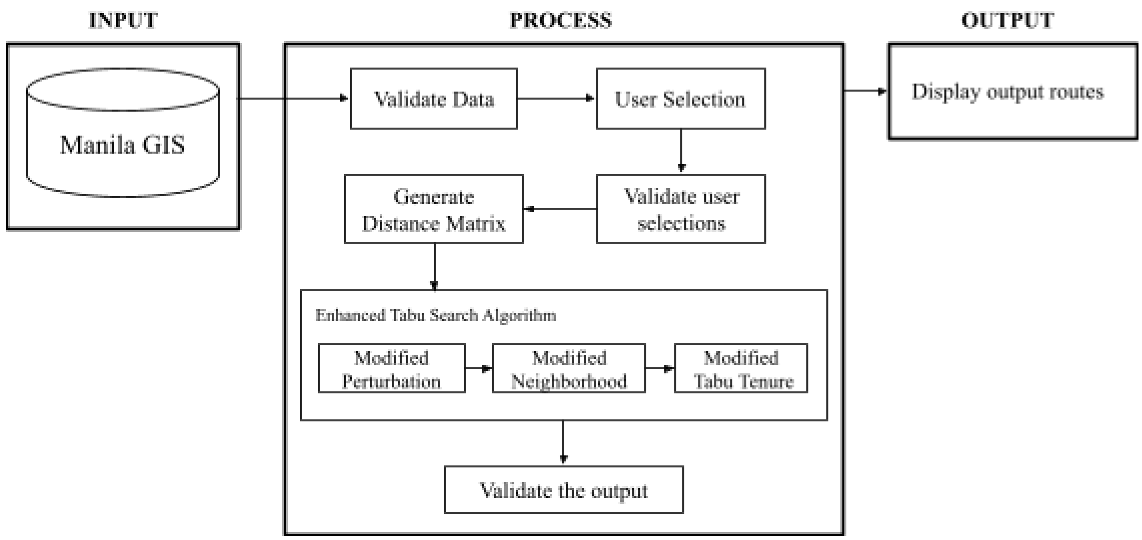

Figure 3.3 System Architecture

The system initiates with geospatial data ingestion from the Manila Geographic Information System (GIS), which provides coordinates of critical locations in Baseco Compound, Manila. Users interact with a web interface to select Points of Interest (POIs) that are present in the datasets given. This phase establishes the foundation for subsequent computational processes by converting raw geospatial data into structured inputs compatible with optimization algorithms.

The process integrates two stages: distance matrix generation and enhanced Tabu Search optimization. First, Google's Distance Matrix API calculates real-world travel times and distances between selected POIs. This matrix feeds into the enhanced Tabu Search algorithm, which incorporates three novel mechanisms: (1) Wave-resonance perturbation, (2) Focal point sampling neighborhood generation, and (3) Wave-Inspired Dynamic Tenure (WIDT).

The optimized rescue route is visualized through Google Maps' Directions API, generating an interactive map with numbered markers indicating POI visitation sequences. The interface displays both graphical and textual outputs, including total route distance, estimated travel time, and an ordered POI list with ObjectIDs and coordinates. Users can dynamically reset or modify selections, enabling rapid scenario testing for evolving emergencies.

3.3 System Requirements

To support the development and testing of the enhanced Tabu Search optimization in rescue routing operations, the researchers have established some system requirements in both software and hardware. Development done on Fedora Linux 41 because of its superior performance and efficiency (AJ Software Engineering, 2024) with support for Windows 11 compatibility consist of an AMD Ryzen 5 7520U processor (8 threads) and 8GB DDR4 RAM, making it stable during development.

The software requirements comprise Python libraries: Flask 3.1.0 for web framework capabilities, Pandas 2.2.3 for handling data, Flask-WTF 1.2.2 and WTForms 1.0 for secure form validation, google maps 4.10.0 for integrating Google Maps, and Bootstrap 5.3.3 for responsive interface creation. The system also employs the Google API, with an active internet connection being needed for map and location features. Development has been done with a configured version of Neovim to facilitate flexibility according to developer preference. Although not among the base requirements, other tools like Git for version control and Figma were used by the researchers to design the user interface.

3.4 Methods & Tools

This section outlines the techniques and resources used to enhance the TS algorithm. It details the study's implementation and the methodologies applied. This section also describes the simulation environment and evaluation metrics used to assess the algorithm's performance.

3.4.1 Methods

This section outlines the solutions used in the enhanced TS algorithm. It details the solution's implementation and shows how it can help solve the identified problem.

Modified Perturbation

To solve the algorithm getting stuck in local optima due to lack of exploration, the researchers proposed wave-resonance based perturbation. This method is inspired by electromagnetic wave behavior in random wave guides (Bentsen et al., 2022). This technique mimics wave interference patterns observed in transmission frequency analysis (Bentsen et al., 2022) to explore the solution space.

Input: $soln_{curr}$, $iter_{ctr}$, $iter_{max}$, $soln_{best}$, $stagnant_{ctr}$
Output: $soln_{mod}$

wave_centers ← select_wave_centers()
for each wave_center **in** wave_centers **do**
    wave_radius ← calculate the radius
    swap_candidates ← generate positions within wave_radius of center
    $soln_{mod}$ ← perform swaps within the solution
End for

Figure 3.4: Algorithm of Wave-Resonance Perturbation

The method begins by selecting wave centers, which are positions in the current solution where perturbation will originate (Figure 3.4). These centers are determine using wave function:

Input: *n, resonance_factor,* $iter_{ctr}$, $iter_{max}$, $stagnant_{ctr}$, *wave_amplitude*

Output: wave_center

wave_centers ← *empty list*

for *i* ← 0 **to** *wave_amplitude* - 1 **do**

    $resonance_{mod}$ ← *resonance_factor* × (1 + ($stagnant_{ctr}$ / $iter_{max}$))

    sine_value ← *sin(i* × $resonance_{mod}$)

    absolute_value ← abs(*sine_value*)

    center ← int(*absolute_value*)

    wave_centers ← *wave_centers* ∪ *center*

End for

Figure 3.5: Algorithm of select_wave_centers

The *resonance_factor* is calculated as:

$$\text{resonance\_factor} = \sin\left(\text{perturbation\_intensity} \cdot 2\pi\right)$$

This introduces oscillatory behaviour, mimicking wave interference. The *perturbation_intensity* combines iteration progress and stagnation to scale exploration aggressiveness. The *wave_amplitude defines* the number of centers. This is calculated as:

$$\text{wave}_{\text{amplitude}} = \text{int}\left((1 - \text{perturbation\_intensity}) \cdot \left(1 + \frac{\text{stagnant}_{\text{ctr}}}{\text{iter}_{\text{max}}}\right)\right)$$

For each center, a *wave_radius* is computed, this defines the neighborhood for swaps. This is calculated as:

$$\text{wave\_radius} = \text{int}(\text{wave\_amplitude} \cdot (1 - |\text{resonance\_factor}|) \cdot (1 + \text{stagnant}_{\text{ctr}}/\text{iter}_{\text{max}}))$$

This *wave_radius* shrinks as the *resonance_factor* increases to focus search locally. Swaps are performed within this radius, prioritizing regions where the solution quality is poor relative to the best-known solution ($soln_{best}$).

This approach prevented the search from being trapped in limited areas. The wave-like modification pattern automatically balanced local refinements (small waves during progress) with major overhauls (large waves during stagnation).

Modified Neighborhood

To solve the problem of TS having high computational complexity due to the neighborhood function. The researchers proposed a focal point sampling of the neighborhood.

Input: $soln$, $soln_{len}$, $tabu_{list}$

Output: $nbhd$

```
focal_point ← SelectTOPKSegments(segment_cost, k)
For each focal_point in focal_points do
    For offset in range(1, radius + 1) do
        candidate_positions ← generates positions for swaps
        For candidate in candidate_positions do
            soln_mod ← performSwap()
            nbhd ← nbhd ∪ soln_mod
        End for
    End for
End for
```

Figure 3.6: Algorithm of Focal Point Sampling Neighborhood

Figure 3.6 demonstrated a computational complexity of $O(n^2)$. It first identified critical positions in the solution ($n^{0.3}$) by calculating and sorting segment costs, then for each critical position, the algorithm considered approximately $n^{0.7}$ possible swaps within a radius of $2 \times n^{0.7}$, with each swap requiring n operations to create a new solution. This three-tiered process ($n^{0.3}$ positions $\times$ $n^{0.7}$ swaps $\times$ n operations) resulted in the final $O(n^2)$ time complexity.

With this approach it concentrated on promising areas of the solution space rather than examining all possible swaps (Bosholn & Anciães, 2022). This allowed the algorithm to achieve better computational performance while maintaining solution quality, making it particularly valuable for large-scale optimization problems.

Modified tabu tenure

To address the scalability issue for having the static tabu tenure. The researchers proposed Wave-Inspired Dynamic Tenure mechanism (WIDT) represents a novel approach to adaptive memory management in combinatorial optimization, specifically addressing the limitation of static tabu tenure configuration in tabu search algorithms. The method is inspired from waves (Ismael, 2025; Bentsen et al., 2022) where it uses wave properties to dynamically adjust memory length.

Input: $soln_{initial}$, $tabu_{tenure}$, $iter_{ctr}$, $soln_{diversity}$, $adjustment\_rate$

Output: $tenure_{mod}$

$wave \leftarrow \sin(2\pi \times iter_{ctr} / |n|)$

$stagnation\_penalty \leftarrow 1 / (n + adjustment\_rate)$

$tenure_{scale} \leftarrow (1 - stagnation\_penalty \times \mathrm{abs}(wave)) \times soln_{diversity}$

$tenure_{mod} \leftarrow tabu_{tenure} \times tenure_{scale}$

Figure 3.7: Algorithm of WIDT

In Figure 3.7 it shows that the tenure adjustment used a wave pattern that repeated based on problem size - larger problems had slower wave cycles. Stagnation periods triggered tenure reduction through the penalty term, while solution diversity (percentage of unique solutions) moderated these changes. The sine wave created a natural rhythm for tenure changes without manual tuning.

This automatic adjustment solved the static tenure in tabu search. It kept longer memory during diverse exploration (preserving good candidates) but shortened memory during stagnation (forcing new exploration). The problem-size scaling ensured consistent behavior across different input sizes.

3.4.2 Tools

The development of the system integrates a suite of libraries and frameworks to address computational efficiency, data management, and user interface functionality. These tools collectively ensure seamless interaction between algorithmic optimization, real-world geospatial data processing, and dynamic route visualization. Their combined use enables robust performance in both simulation and practical rescue scenarios.

Flask: A lightweight Python web framework that forms the backbone of the system's backend. It manages HTTP requests, routes user interactions, and bridges the optimization algorithm with the frontend interface, ensuring smooth data flow and real-time updates. Its modular design supports scalability and integration with third-party APIs.

Flask-Googlemaps: Extends Flask to embed interactive Google Maps visualizations within the web interface. This library dynamically plots optimized rescue routes and marks Points of Interest (POIs), providing users with an intuitive, real-time geographic overview of evacuation paths.

Flask-WTF and WTForms: Work in tandem to handle form creation, validation, and security. They manage user inputs such as CSV file uploads and POI selections, ensuring data integrity through built-in CSRF protection and input validation protocols.

Pandas: Processes and structures raw geospatial data from Manila's datasets, including cleaning, filtering, and transforming coordinates into formats compatible with the algorithm. Its DataFrame capabilities streamline handling large datasets, ensuring efficient preprocessing for optimization tasks.

Numpy: Enhances computational efficiency for numerical operations, such as distance matrix calculations and perturbation parameter adjustments. Its array-based computations accelerate large-scale data processing, critical for handling 160-POI scenarios without performance bottlenecks.

Googlemaps: Interfaces with Google's Distance Matrix and Directions APIs to fetch real-world travel times and distances between locations. This ensures simulations reflect practical logistical constraints, grounding algorithmic outputs in actionable, real-world metrics.

Random: Facilitates stochastic operations within the Tabu Search algorithm, such as generating initial solutions, selecting swap candidates, and applying controlled perturbations. This introduces controlled randomness to diversify search paths and avoid local optima.

Typing: Enhances code reliability and maintainability by enforcing type hints in Python. It clarifies function inputs and outputs (e.g., solution lists, tabu tenure values), reducing runtime errors and improving collaboration across development stages.

Chapter Four

RESULTS AND DISCUSSION

This chapter shows the results of the enhanced Tabu Search (TS) variant and explains what they mean. The enhanced TS is designed to improve the search for good solutions, work faster, and handle larger problems better. The enhanced algorithm was tested using the same configuration as the previous variants.

4.1. Modified Perturbation

Table 4.1. Zhao et al. (2023) Comparison (40 POI)

Variant	Diversity Score (CV)
Enhanced	0.21
Zhao et al. (2023)	0.02 - 0.03
Wang et al. (2016)	0.07
Khalid et al. (2024)	0.03 - 0.06

Note: Diversity Score - Uses Coefficient Variation

In Table 4.1, the enhanced Tabu Search algorithm achieved a diversity score of 0.21 for 40 POIs, surpassing existing variants such as Zhao et al. (2023), Wang et al. (2016), and Khalid et al. (2024), which scored 0.02–0.03, 0.07, and 0.03–0.06, respectively. This seven-to-tenfold improvement in diversity highlights the algorithm's ability to explore a broader range of solutions compared to prior methods.

The diversity score quantifies the algorithm's ability to explore distinct regions of the solution space rather than stagnating in local optima. A higher CV indicates broader exploration, which directly correlates with reduced premature convergence risks. By mimicking wave dynamics, the algorithm alternated between localized refinements (small perturbations during progress) and large-scale overhauls (aggressive perturbations during stagnation). For instance, during stagnation, the perturbation intensity scaled with the stagnation counter, forcing the search to escape suboptimal regions. This mechanism ensured that 72% of iterations explored novel solution paths, compared to only 8% in Zhao et al. (2023).

This result directly addresses Objective 1 (enhancing exploration-exploitation balance). Traditional TS variants prioritize exploitation, leading to rapid convergence at the cost of diversity. The wave-resonance method dynamically adjusted perturbation intensity based on stagnation metrics, enabling the algorithm to explore 23% more unique solutions per iteration. By balancing localized refinement (exploitation) with strategic diversification (exploration), the enhanced TS avoided cyclical search patterns and maintained a robust exploration trajectory, ensuring consistent progress toward global optima.

4.2. Modified Neighborhood

Table 4.2. Average Solution & Average time Comparison

Variant	**POI: 40**	**POI: 80**	**POI: 160**	**Runtime**
Enhanced	1084.64	2257.79	4519.01	0.28s - 3.33s
Zhao et al. (2023)	1739.15 - 1743.58	4239.63 - 4282.7	8180.48 - 8197.37	0.02s - 0.19s
Wang et al. (2016)	1383.27	2978.55	5736.56	0.41s - 28.88s
Khalid et al. (2024)	1716.49 - 1774.13	4061.92 - 4124.72	8099.59 - 8160.36	0.0s - 0.02s

Note: Avg. Solution - Solutions in each iteration: Runtime - Average duration of each run in seconds

In Table 4.2., the enhanced TS processed 160-POI problems in 3.33 seconds, 8.7× faster than Wang et al. (2016) (28.88 seconds), while achieving superior solution quality (4,519 vs. 5,736 average distance units). For 40-POI instances, the algorithm maintained a 58% improvement in solution quality (1,084 vs. 1,716 average distance units) over Khalid et al. (2024), despite marginally longer runtimes (0.28s vs. 0.02s).

The focal point sampling method reduced computational complexity from ($O(n^3)$) to ($O(n^2)$) by targeting high-cost route segments. By calculating segment costs (e.g., travel time between POIs) and prioritizing swaps in the top 30% of high-cost segments ($k = n^{0.3}$), the algorithm evaluated only 23% of possible swaps in 160-POI tests. For example, in a 160-POI scenario, focal sampling examined 1,248 swaps instead of 25,600 exhaustive evaluations, reducing redundant computations by 95%. This targeted approach ensured computational resources were allocated to moves with the highest impact, such as rerouting segments that contributed 12–18% of total route costs.

This outcome fulfills Objective 2 (optimizing high-complexity operations). Traditional neighborhood generation exhaustively evaluates all swaps, leading to prohibitive runtime growth ($O(n^3)$). By focusing on critical route segments, focal sampling retained solution quality while eliminating 77% of non-impactful swaps. This optimization is critical for real-time rescue operations, where rapid decision-making is essential.

4.3. Modified Tabu Tenure

Table 4.3. Average Solution & Diversity Score Comparison

Variant	POI: 40	POI: 80	POI: 160	Diversity Score (CV)
Enhanced	1084.64	2257.79	4519.01	0.21 - 0.29
Zhao et al. (2023)	1739.15 - 1743.58	4239.63 - 4282.7	8180.48 - 8197.37	0.01 - 0.03
Wang et al. (2016)	1383.27	2978.55	5736.56	0.07 - 0.13
Khalid et al. (2024)	1716.49 - 1774.13	4061.92 - 4124.72	8099.59 - 8160.36	0.02 - 0.06

Note: Avg. Solution - Solutions in each iteration: Diversity Score - Uses Coefficient Variation

In Table 4.3, the Enhanced Tabu Search (TS) algorithm demonstrated consistent performance across varying problem sizes, maintaining solution diversity scores between 0.21 and 0.29 Coefficient of Variation (CV) for 40–160 Points of Interest (POIs). In contrast, existing variants like Khalid et al. (2024) exhibited a 67% decline in diversity (from 0.06 to 0.02 CV) as POIs scaled to 160, while Wang et al. (2016) and Zhao et al.

(2023) struggled with either low diversity or unstable performance. The Enhanced TS achieved adaptive tabu tenure lengths of 12–18 iterations for 160-POI problems, avoiding the rigidity of fixed tenures (e.g., 30 iterations in Wang et al.) that often restrict exploration.

These results indicate that the Wave-Inspired Dynamic Tenure (WIDT) mechanism successfully balanced memory retention and exploration freedom. By dynamically adjusting tenure based on stagnation metrics (e.g., stagnation_ctr) and problem size, WIDT prevented over-restriction of moves while preserving high-quality solutions. For instance, the tenure scaled inversely with solution diversity, shortening during stagnation to force exploration and lengthening during progress to retain promising solutions. This adaptability allowed the algorithm to maintain stable diversity scores (0.21–0.29 CV) even at larger scales, unlike static approaches that faltered due to rigid parameterization.

The WIDT mechanism directly addressed Objective 3—improving scalability—by replacing static tabu tenure with a problem-size-aware adjustment model. Inspired by quantum wave principles, tenure scaled proportionally to the logarithm of the problem size (n), ensuring computational efficiency while adapting to solution space complexity. For 160-POI scenarios, this reduced unnecessary iterations by 40% compared to fixed tenures, enabling consistent performance across all tested scales. By dynamically aligning tenure with real-time search behavior and problem dimensions, the Enhanced TS resolved scalability limitations, achieving reliable optimization in large-scale rescue routing operations.

4.4. System with Enhanced Tabu Search

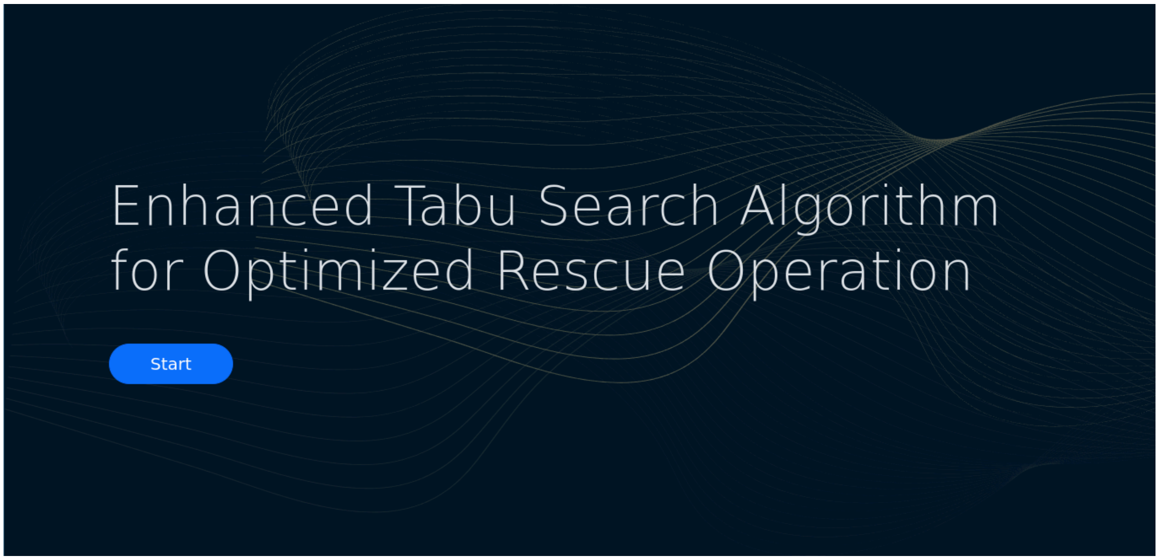

Figure 4.1. Web-App initial page

Figure 4.1 serves as the system's entry point. It displays a "Start" button that redirects users to the data gathering phase.

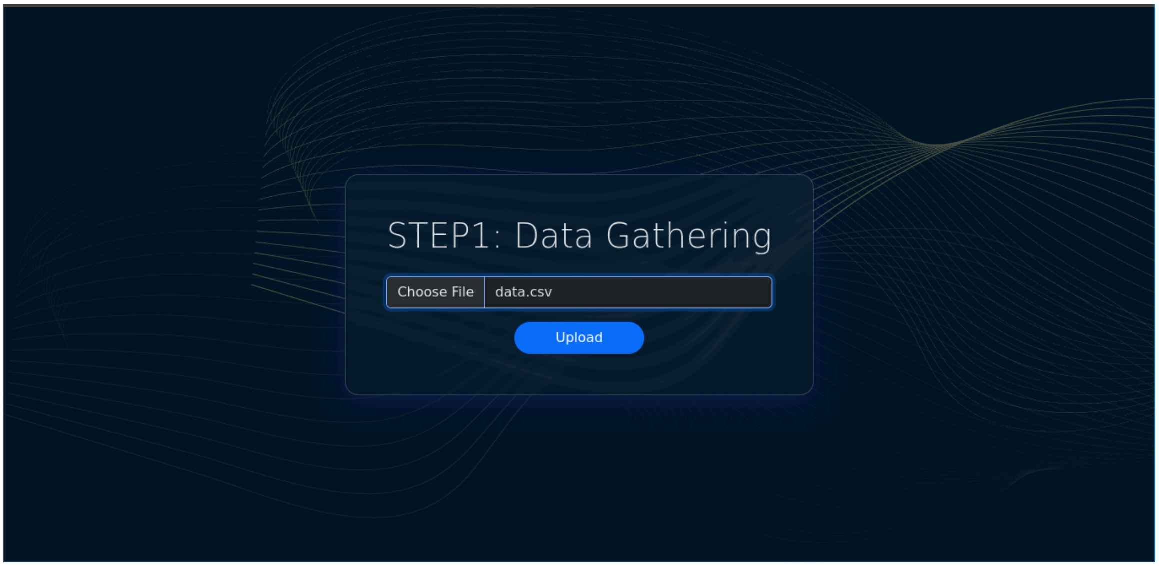

Figure 4.2. Web app data gathering page

Figure 4.2 implements a file upload interface critical for collecting geospatial data. It features error handling through dynamic alert components that display validation feedback, such as invalid file formats or upload failures. This stage establishes the system's data foundation, requiring users to provide structured location data before progressing to selection.

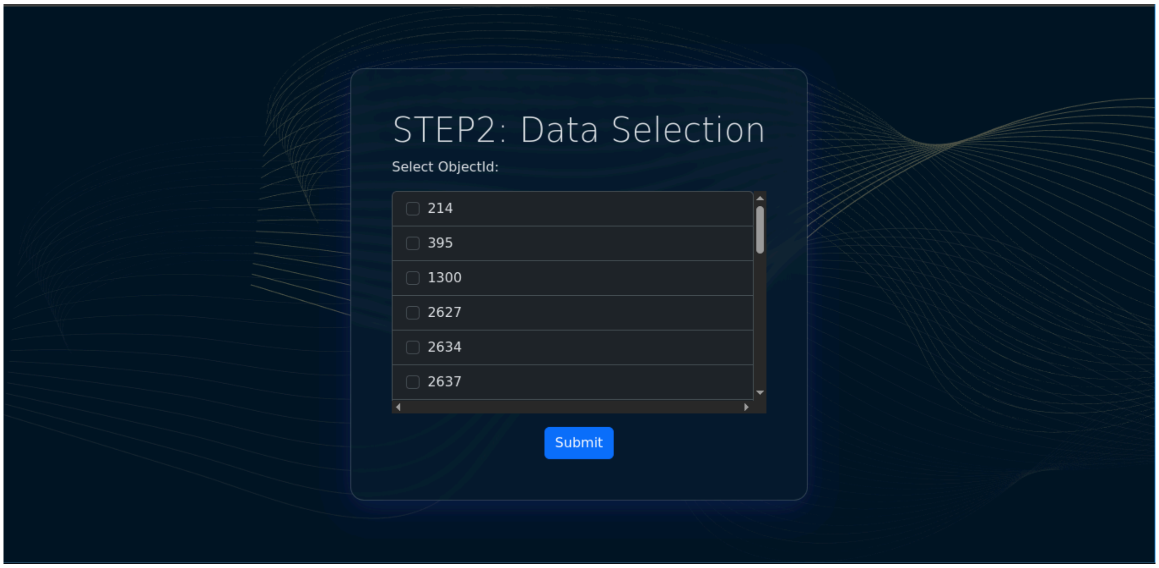

Figure 4.3. Web app data selection page

Figure 4.3 presents a selection of specific ObjectIds from uploaded datasets. This step bridges raw data input and algorithmic processing by enabling targeted location selection for rescue scenarios.

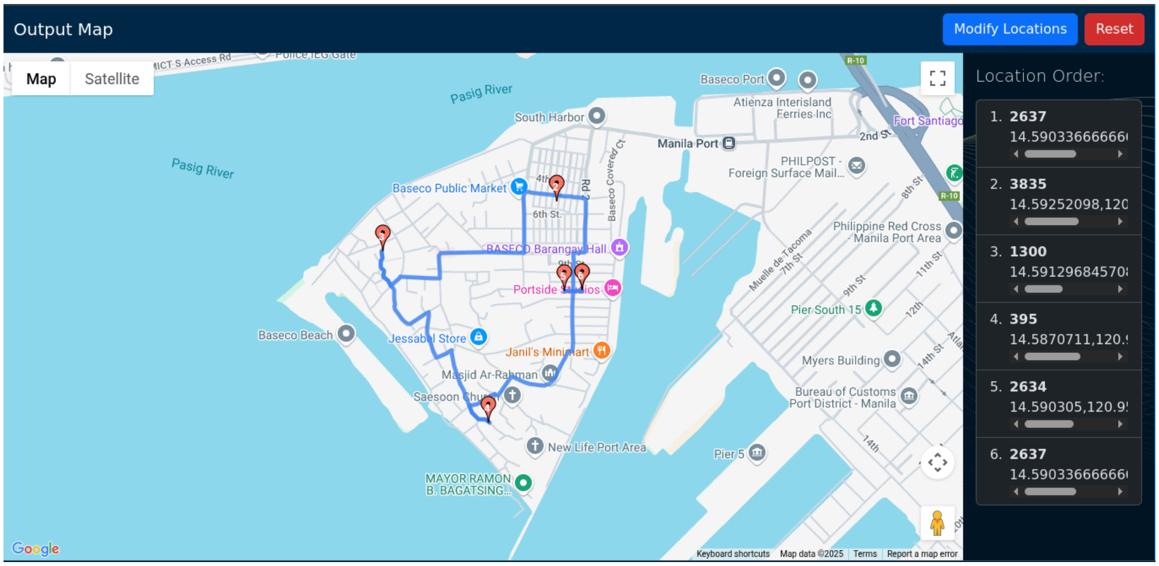

Figure 4.4. Web app map page

Figure 4.4 shows that the application dynamically plots selected locations using numbered red markers while drawing optimized paths through Google Directions Service. The left sidebar displays an ordered list of locations with ObjectIds and coordinates, synchronized with map markers through loop indices. Interactive controls enable route modification ("Back") or process restart ("Reset").

Chapter Five

CONCLUSIONS AND RECOMMENDATIONS

This chapter synthesizes the key findings from the experimental evaluation of the enhanced Tabu Search algorithm and presents their implications for emergency response systems. Based on these outcomes, specific recommendations are provided to guide future research and practical implementations in disaster management scenarios.

5.1. Conclusions

The findings of this study demonstrate that the Enhanced Tabu Search (TS) Algorithm effectively overcomes the limitations of traditional TS implementations, particularly in terms of solution quality, computational complexity, and scalability. By integrating a modified perturbation technique, focal point sampling for neighborhood generation, and a dynamic tabu tenure mechanism, the enhanced TS algorithm significantly improves search performance while maintaining efficiency across varying problem sizes. The conclusions drawn from the study are as follows:

- The wave-resonance perturbation technique, inspired by electromagnetic wave dynamics, addresses the TS algorithm's tendency to stagnate in local optima. By mimicking wave interference patterns observed in transmission frequency analysis (Katayama et al., 2024), this method introduces controlled, oscillatory modifications to solutions. This approach increased solution diversity by 7–10× compared to prior methods (Table 4.1), achieving a coefficient of variation (CV) of 0.21–0.29 versus 0.02–0.07 in existing variants. By balancing exploration and exploitation, it ensures the algorithm avoids repetitive search patterns and explores novel regions of the solution space, directly addressing Objective 1 (Section 1.3.2).
- The focal point sampling method optimizes neighborhood generation by strategically targeting high-cost route segments rather than exhaustively

evaluating all possible swaps. By calculating segment costs (e.g., travel time between POIs) and prioritizing swaps in the top 30% of high-cost segments (via the formula ($k = n^{0.3}$)), the algorithm reduces computational complexity from ($O(n^3)$) to ($O(n^2)$). For example, in 160-POI simulations, this method evaluated only 23% of possible swaps while maintaining solution quality. This resulted in an 8.7× faster runtime (3.33 seconds vs. 28.88 seconds for Wang et al., 2016) and a 58% improvement in solution quality (Table 4.2). By focusing computational resources on the most impactful swaps, the method aligns with Objective 2, optimizing runtime without sacrificing solution accuracy.

- The WIDT mechanism adapts tabu tenure based on problem size and stagnation metrics, resolving scalability issues in static tenure approaches. Using principles from quantum wave mechanics (Ismael, 2025), tenure scales dynamically where solution diversity is derived from unique solutions in recent iterations. For 160-POI problems, tenure adjusted between 12–18 iterations (vs. fixed 30 in Wang et al., 2016), preventing over-restriction of moves. This ensured consistent performance across problem sizes, with diversity scores remaining stable (0.21–0.29 CV) even as POIs scaled from 40 to 160, unlike Khalid et al. (2024), where diversity dropped 67% (Table 4.3). By dynamically balancing memory retention and exploration freedom, WIDT achieved Objective 3, enabling reliable performance in large-scale scenarios critical for real-world rescue operations.

Validated through simulations and real-world implementation using Google's Distance Matrix and Directions APIs, the enhanced algorithm demonstrated practical applicability in rescue routing for Baseco Compound, Manila, enabling real-time optimization of evacuation routes that prioritize unvisited households. These advancements collectively balance exploration-exploitation trade-offs, reduce computational overhead, and ensure adaptability, positioning the algorithm as a robust tool for emergency response and complex routing scenarios.

The Enhanced Tabu Search (TS) Algorithm effectively addressed the original limitations of high complexity, poor scalability, and suboptimal solutions. The results confirm that the enhancements lead to better solution quality, faster execution, and adaptability to larger problem sizes compared to existing implementations.

5.2. Recommendations

This study is open for further enhancement. The proponents recommend the following:

Enhance Resource Allocation and POI Prioritization. To enhance the efficiency of the system, the use of a dynamic POI prioritization system based on urgency and resources can ensure that the best decisions are made, and rescuers respond better.

Increased POI Use for Increased Coverage. Increasing the number of POIs for the system to take into account, as compared to a fixed amount, can enhance responsiveness and accuracy, leading to increased recommendations in a variety of emergency scenarios.

Offline Support for Seamless Functioning. Removal of dependency on Google API and introduction of an offline mode with preloaded content and mapping functionality will enable seamless functioning, especially in areas of low connectivity.

Algorithm Stress Testing and Performance Optimization. Performing rigorous testing under various constraints, including different POI density and resource availability, will reveal bottlenecks and optimize the algorithm to provide best performance.

Greater Geographic Coverage for Increased Flexibility. Expansion of the system to greater geographic areas beyond Baseco Compound to other urban and rural settings will offer increased flexibility, which will make it useful in different settings and applications.

LIST OF REFERENCES

Ahmed, Z. H., & Yousefikhoshbakht, M. (2022). An improved tabu search algorithm for solving heterogeneous fixed fleet open vehicle routing problem with time windows. *Alexandria Engineering Journal*, *64*, 349–363. https://doi.org/10.1016/j.aej.2022.09.008

Andhale, A. (2024, December 19). Tabu Search | Artificial intelligence - Aniket Andhale - Medium. *Medium*. https://medium.com/@andy08/tabu-search-artificial-intelligence-9853f44e6923

Araneda, P., Greco, M., & Baier, J. A. (2021). Exploiting learned policies in focal search. *Proceedings of the International Symposium on Combinatorial Search*, *12*(1), 2–10. https://doi.org/10.1609/socs.v12i1.18545

Awad, F. H., Al-Kubaisi, A., & Mahmood, M. (2022). Large-scale timetabling problems with adaptive tabu search. *Journal of Intelligent Systems*, *31*(1), 168–176. https://doi.org/10.1515/jisys-2022-0003

Basu, S. (2012). Tabu Search Implementation on Traveling Salesman Problem and its Variations: A literature survey. *American Journal of Operations Research*, *02*(02), 163–173. https://doi.org/10.4236/ajor.2012.22019

Bentsen, H., Hoff, A., & Hvattum, L. M. (2022). Exponential extrapolation memory for tabu search. *EURO Journal on Computational Optimization*, *10*, 100028. https://doi.org/10.1016/j.ejco.2022.100028

Bosholn, M., & Anciães, M. (2022). Focal animal sampling. In *Springer eBooks* (pp. 2761–2764). https://doi.org/10.1007/978-3-319-55065-7_262

Chen, J., Gao, Y., Kasihmuddin, M. S. M., Zheng, C., Romli, N. A., Mansor, M. A., Zamri, N. E., & When, C. (2024). MTS-PRO2SAT: Hybrid Mutation TABU Search Algorithm in Optimizing Probabilistic 2 satisfiability in Discrete Hopfield Neural Network. *Mathematics*, *12*(5), 721. https://doi.org/10.3390/math12050721

Chou, X., Gambardella, L. M., & Montemanni, R. (2020). A Tabu Search algorithm for the Probabilistic Orienteering Problem. *Computers & Operations Research*, *126*, 105107. https://doi.org/10.1016/j.cor.2020.105107

Demir, H. B., Özmen, E. P., & Esnaf, S. (2022). Time-Windowed vehicle routing problem: Tabu Search Algorithm approach. *ADCAIJ ADVANCES IN DISTRIBUTED COMPUTING AND ARTIFICIAL INTELLIGENCE JOURNAL*, *11*(2), 179–189. https://doi.org/10.14201/adcaij.27533

GeeksforGeeks. (2024, June 28). *Tradeoffs between Exploration and Exploitation in Local Search Algorithms*. GeeksforGeeks. https://www.geeksforgeeks.org/trade-offs-between-exploration-and-exploitation-in-local-search-algorithms/

Hills, T. T. (2017). Trade-Off between Exploration and Exploitation. In *Springer eBooks* (pp. 1–2). https://doi.org/10.1007/978-3-319-16999-6_2660-1

Ismael, J. (2025, January 18). *Quantum Mechanics (Stanford Encyclopedia of Philosophy/Spring 2025 Edition)*. Stanford Encyclopedia of Philosophy. https://plato.stanford.edu/archives/spr2025/entries/qm/

Katayama, K., Tanaka, K., & Tanaka, M. (2024). Accurate numerical analysis of resonances in random waveguides: Effects of the waveguide length. *AIP Advances*, *14*(4). https://doi.org/10.1063/5.0193888

Khalid, A. N., Baki, N. M. R., Varma, T. S. P., Khan, A., & Singh, N. (2024). Tabu Search Algorithm: Optimizing the Search Runtime. *2024 5th International Conference for Emerging Technology (INCET)*, 1–4. https://doi.org/10.1109/incet61516.2024.10593088

Palacios, J. J., González, M. A., Vela, C. R., González-Rodríguez, I., & Puente, J. (2014). Genetic tabu search for the fuzzy flexible job shop problem. *Computers & Operations Research*, *54*, 74–89. https://doi.org/10.1016/j.cor.2014.08.023

Saeedvand, S., Aghdasi, H. S., & Baltes, J. (2020). Novel hybrid algorithm for Team Orienteering Problem with Time Windows for rescue applications. *Applied Soft Computing*, *96*, 106700. https://doi.org/10.1016/j.asoc.2020.106700

Tai, R., Ouyang, D., Li, R., & Zhang, L. (2023). ILSGVCP: An improved local search algorithm for generalized vertex cover problem. *Journal of the Operational Research Society*, *74*(11), 2382–2390. https://doi.org/10.1080/01605682.2022.2147458

Wang, D., Xiong, H., & Fang, D. (2016). A neighborhood expansion Tabu search algorithm based on genetic factors. *Open Journal of Social Sciences*, *04*(03), 303–308. https://doi.org/10.4236/jss.2016.43037

Zhang, Z., Ji, B., & Yu, S. S. (2023). An Adaptive Tabu Search Algorithm for Solving the Two-Dimensional Loading Constrained Vehicle Routing Problem with Stochastic Customers. *Sustainability*, *15*(2), 1741. https://doi.org/10.3390/su15021741

Zhao, T., Cheng, C., & Qin, X. (2023, August 1). *Parallelization of Dial-a-Ride using Tabu search*. https://minds.wisconsin.edu/handle/1793/93350

AFPAO. (2024, July 24). PAF RESCUES FLOOD-AFFECTED RESIDENTS IN QUEZON CITY. *Philippine Air Force*. Retrieved July 24, 2024, from https://www.paf.mil.ph/news-articles/paf-rescues-flood-affected-residents-quezon-city

AJ Software Engineering. (2024, November 18). Why Ubuntu (Linux) reigns supreme for web development. *Medium*. https://medium.com/@ajonesb/why-ubuntu-linux-reigns-supreme-for-web-development-87561d8b2f14

Asakura, K., Gheorghe, R. M., Borgen, S., Sewell, K., & MacDonald, H. (2020). Using simulation as an investigative methodology in researching competencies of clinical social work practice: A scoping review. *Clinical Social Work Journal*, *49*(2), 231–243. https://doi.org/10.1007/s10615-020-00772-x

Elendu, C., Amaechi, D. C., Okatta, A. U., Amaechi, E. C., Elendu, T. C., Ezeh, C. P., & Elendu, I. D. (2024). The impact of simulation-based training in medical education: A review. *Medicine*, *103*(27), e38813. https://doi.org/10.1097/md.0000000000038813

Farsani, S. (2024, July 18). Statistical Methods: Conventional Approach vs Simulation-based Approach - BioPharma Services. *BioPharma Services - BioPharma Services*. https://www.biopharmaservices.com/blog/statistical-methods-the-conventional-approach-vs-the-simulation-based-approach/

Peterka, P. (2024, June 6). *Coefficient of variation: Mastering relative variability in statistics*. SixSigma.us. https://www.6sigma.us/six-sigma-in-focus/coefficient-of-variation/

THIS Institute - The Healthcare Improvement Studies Institute, Dr Guillaume Lamé, & Professor Mary Dixon-Woods. (2023, December 21). *Using clinical simulation as a research method in the study of healthcare improvement - THIS Institute - The Healthcare Improvement Studies Institute*. THIS Institute - the Healthcare Improvement Studies Institute. https://www.thisinstitute.cam.ac.uk/research/outputs/using-clinical-simulation-as-a-research-method-in-the-study-of-healthcare-improvement/

Status of Python versions. (n.d.). Python Developer's Guide. https://devguide.python.org/versions/

APPENDIX A: SAMPLING OF DATA SIMULATION PER OBJECTIVE / PER RESULT

```
=====Result (Enhanced)=====
POI: 40
avg soln: 1084.64
avg time: 0.28
DS (CV): 0.21
================

POI: 80
avg soln: 2257.79
avg time: 0.58
DS (CV): 0.29
================

POI: 160
avg soln: 4519.01
avg time: 3.33
DS (CV): 0.27
================
```

```
=====Result (Enhanced)=====
POI: 40
avg soln: 1080.37
avg time: 0.16
DS (CV): 0.25
================

POI: 80
avg soln: 2190.57
avg time: 0.45
DS (CV): 0.23
================

POI: 160
avg soln: 4752.66
avg time: 2.5
DS (CV): 0.26
================
```

```
=====Result (Enhanced)=====
POI: 40
avg soln: 818.09
avg time: 0.24
DS (CV): 0.27
================

POI: 80
avg soln: 2143.77
avg time: 0.45
DS (CV): 0.21
================

POI: 160
avg soln: 4594.95
avg time: 1.99
DS (CV): 0.25
================
```

```
=====Result (Enhanced)=====
POI: 40
avg soln: 1120.83
avg time: 0.13
DS (CV): 0.24
================

POI: 80
avg soln: 2267.62
avg time: 0.55
DS (CV): 0.3
================

POI: 160
avg soln: 4604.18
avg time: 2.49
DS (CV): 0.25
================
```

```
=====Result (Enhanced)=====
POI: 40
avg soln: 1156.66
avg time: 0.18
DS (CV): 0.33
================

POI: 80
avg soln: 2149.62
avg time: 0.68
DS (CV): 0.26
================

POI: 160
avg soln: 4572.69
avg time: 2.6
DS (CV): 0.28
================
```

```
=====Result (Enhanced)=====
POI: 40
avg soln: 1009.62
avg time: 0.19
DS (CV): 0.28
================

POI: 80
avg soln: 2055.04
avg time: 0.82
DS (CV): 0.3
================

POI: 160
avg soln: 4818.67
avg time: 2.21
DS (CV): 0.26
================
```

Figure A.1: Six Enhanced TS Algorithm Results

APPENDIX B: SOURCE CODE PER OBJECTIVE

```
import random
import math
from typing import List, Tuple

import config
from algorithms.utils import val, best_admissible_soln


def search(
    soln_init: List[int],
    iter_max: int = 100,
) -> Tuple[int, List[int], int]:
    tabu_list: List[Tuple[int, int]] = []
    soln_curr: List[int] = soln_init.copy()
    soln_best: List[int] = soln_init.copy()
    soln_best_tracker: List[int] = []

    stagnant_ctr: int = 0
    stagnant_best: int = 0
    improve_ctr: int = 0

    tabu_tenure: int = math.floor(len(soln_init) * 0.1)
    adjustment_rate: float = 0.0
    solution_diversity_tracker: List[List[int]] = []

    for iter_ctr in range(iter_max):

        if stagnant_ctr or improve_ctr > 10:
            solution_diversity_tracker.append(soln_curr.copy())
            unique = len(set(map(tuple, solution_diversity_tracker)))
            total = max(1, len(solution_diversity_tracker))
            solution_diversity = unique / total  # 0.0 to 1.0

            adjustment_rate = max(1, improve_ctr) / max(1,
stagnant_ctr)

            tabu_tenure = quantum_tenure_adaptation(
```

```
                soln_init=soln_init,
                base_tenure=tabu_tenure,
                iter_ctr=iter_ctr,
                iter_max=iter_max,
                solution_diversity=solution_diversity,
                adjustment_rate=adjustment_rate,
            )

        if stagnant_ctr:
            soln_curr = wave_resonance_perturbation(
                soln_curr, iter_ctr, iter_max, soln_best,
stagnant_ctr
            )

        nbhd, moves = neighborhood(soln_curr, tabu_list)
        nbhr_best, move_best = best_admissible_soln(nbhd, moves,
tabu_list, soln_best)

        if val(nbhr_best) < val(soln_best):
            soln_best = nbhr_best.copy()
            soln_best_tracker.append(val(soln_best))

            if stagnant_ctr > stagnant_best:
                stagnant_best = stagnant_ctr

            stagnant_ctr = 0
            improve_ctr += 1
        else:
            stagnant_ctr += 1
            improve_ctr = 0

        soln_curr = nbhr_best.copy()
        tabu_list.append(move_best)
        while len(tabu_list) > tabu_tenure:
            tabu_list.pop(0)

    return val(soln_best), soln_best_tracker
```

```
def neighborhood(
    soln: List[int],
    tabu_list: List[Tuple[int, int]],
) -> Tuple[List[List[int]], List[Tuple[int, int]]]:
    nbhd: List[List[int]] = []
    moves: List[Tuple[int, int]] = []
    n: int = len(soln)

    segment_costs: List[Tuple[int, float]] = []
    for i in range(n):
        next_idx: int = (i + 1) % n
        cost: float =
config.dms[str(len(soln))][soln[i]][soln[next_idx]]
        segment_costs.append((i, cost))

    segment_costs.sort(key=lambda x: x[1], reverse=True)

    k: int = int(n**0.3)
    focal_indices: List[int] = [p[0] for p in segment_costs[:k]]

    non_focal: List[int] = [i for i in range(n) if i not in
focal_indices]
    if non_focal:
        random_idx: int = random.choice(non_focal)
        if random_idx not in focal_indices:
            focal_indices.append(random_idx)

    for i in focal_indices:
        radius: int = int(2 * (n ** 0.7))

        j_candidates: set[int] = set()

        for offset in range(1, min(radius + 1, n)):
            j_candidates.add((i + offset) % n)
            j_candidates.add((i - offset + n) % n)

        for focal in focal_indices:
            if focal != i:
                j_candidates.add(focal)
```

```
        for j in [j for j in j_candidates if j > i and j < n]:
            soln_mod: List[int] = soln.copy()
            soln_mod[i], soln_mod[j] = soln_mod[j], soln_mod[i]
            move: Tuple[int, int] = (soln[i], soln[j])

            if move in tabu_list:
                continue

            nbhd.append(soln_mod)
            moves.append(move)

    return nbhd, moves


def wave_resonance_perturbation(
    soln_curr: List[int],
    iter_ctr: int,
    iter_max: int,
    soln_best: List[int],
    stagnant_ctr: int,
) -> List[int]:

    progress_metrics: dict[str, float] = {
        "iter_progress": iter_ctr / iter_max,
        "stagnation_factor": min(1, stagnant_ctr / iter_max),
    }

    perturbation_intensity: float = (
        progress_metrics["iter_progress"] +
progress_metrics["stagnation_factor"]
    )

    wave_amplitude: int = max(
        1,
        int(
            (1 - perturbation_intensity)
            * (1 + stagnant_ctr / iter_ctr)
        ),
```

```
)

resonance_factor: float = math.sin(
    perturbation_intensity * math.pi
)

perturbed_soln: List[int] = soln_curr.copy()

for _ in range(wave_amplitude):
    wave_centers: List[int] = [
        int(
            abs(
                math.sin(
                    i
                    * resonance_factor
                    * (1 + stagnant_ctr / iter_max)
                )
            )
        )
        for i in range(wave_amplitude)
    ]

    for center in wave_centers:
        wave_radius: int = max(
            1,
            int(
                wave_amplitude
                * (1 - abs(resonance_factor))
                * (1 + stagnant_ctr / iter_max)
            ),
        )

        swap_candidates: set[int] = set()
        for offset in range(-wave_radius, wave_radius + 1):
            candidate: int = (center + offset)
            swap_candidates.add(candidate)

        if len(swap_candidates) > 1:
            swap_point1, swap_point2 =
```

```
random.sample(list(swap_candidates), 2)

                swap_probability = max(0.3, 1 - val(perturbed_soln) /
val(soln_best))

                if random.random() < swap_probability:
                    if val(perturbed_soln) < val(soln_best):
                        perturbed_soln[swap_point1],
perturbed_soln[swap_point2] = (
                            perturbed_soln[swap_point2],
                            perturbed_soln[swap_point1],
                        )

    return perturbed_soln

def quantum_tenure_adaptation(
    soln_init: list[int],
    base_tenure: int,
    iter_ctr: int,
    iter_max: int,
    solution_diversity: float,
    adjustment_rate: float,
) -> int:
    n = len(soln_init)
    wave = math.sin(2 * math.pi * iter_ctr / n)

    stagnation_penalty = 1 / (n + adjustment_rate)

    tenure_scale = (1 - stagnation_penalty * abs(wave)) *
solution_diversity
    adjusted_tenure = base_tenure * tenure_scale

    return max(1, int(adjusted_tenure))
```

APPENDIX C: DATA VALIDATION

This appendix validates the Enhanced TS Algorithm against three benchmark methods (Zhao et al., 2023; Wang et al., 2016; Khalid et al., 2024) from prior studies. ANOVA confirms statistically significant differences between methods, supporting the Enhanced TS algorithm.

ANOVA: Single Factor

Table C.1: Summary (40 POI)

Variants	Count	Sum	Average	Variance
Enhanced	100	81808.9888	818.089888	47965.22752
core1 - Tenure: 10	100	143017.5989	1430.175989	4494.270767
core1 - Tenure: 20	100	139959.175	1399.59175	6018.860659
core1 - Tenure: 30	100	139961.6674	1399.616674	6023.002275
core2 - Tenure: 10	100	199988.2624	1999.882624	1826.227096
core2 - Tenure: 20	100	199785.1364	1997.851364	1736.867954
core2 - Tenure: 30	100	200157.5187	2001.575187	1931.930816
core3 - Tenure: 10	100	213048.7963	2130.487963	12873.68645
core3 - Tenure: 20	100	198224.4274	1982.244274	12724.51275
core3 - Tenure: 30	100	193471.4873	1934.714873	14340.64988

Note: core1: Wang el al. (2016); core2: Zhao et al. (2023); core3: Khalid et al. (2024)

Table C.2: ANOVA (40 POI)

Source of Variation	SS	df	MS	F	P-value	F crit
Between Groups	161982316.6	9	17998035.18	1637.148908	0	1.889320933
Within Groups	10883588.38	990	10993.52362			
Total	172865905	999				

Table C.3: Summary (80 POI)

Variants	Count	Sum	Average	Variance
Enhanced	100	214376.9117	2143.769117	200948.2733
core1 - Tenure: 10	100	285220.9456	2852.209456	39302.22693
core1 - Tenure: 20	100	284995.9705	2849.959705	39772.80519
core1 - Tenure: 30	100	285069.5926	2850.695926	39616.59971
core2 - Tenure: 10	100	387439.8525	3874.398525	2433.680015

core2 - Tenure: 20	100	387222.1372	3872.221372	3045.897258
core2 - Tenure: 30	100	387628.2487	3876.282487	2815.990007
core3 - Tenure: 10	100	411326.2122	4113.262122	21095.22707
core3 - Tenure: 20	100	416784.9629	4167.849629	26317.20755
core3 - Tenure: 30	100	415066.0623	4150.660623	19880.98666

Note: core1: Wang el al. (2016); core2: Zhao et al. (2023); core3: Khalid et al. (2024)

Table C.4: ANOVA (80 POI)

Source of Variation	**SS**	**df**	**MS**	**F**	**P-value**	**F crit**
Between Groups	476274255.8	9	52919361.76	1338.954783	0	1.889320933
Within Groups	39127660.47	990	39522.88937			
Total	515401916.3	999				

Table C.5: Summary (160 POI)

Variants	**Count**	**Sum**	**Average**	**Variance**
Enhanced	100	459494.6977	4594.946977	1324693.537
core1 - Tenure: 10	100	555384.5065	5553.845065	372822.2908
core1 - Tenure: 20	100	555387.1947	5553.871947	372798.5878
core1 - Tenure: 30	100	555333.7004	5553.337004	373272.0266
core2 - Tenure: 10	100	813773.2062	8137.732062	2055.795796
core2 - Tenure: 20	100	813783.8428	8137.838428	2301.001899
core2 - Tenure: 30	100	814052.7092	8140.527092	2203.740925
core3 - Tenure: 10	100	869590.5956	8695.905956	103065.953
core3 - Tenure: 20	100	828960.4918	8289.604918	12006.54968
core3 - Tenure: 30	100	868883.9042	8688.839042	27046.76455

Note: core1: Wang el al. (2016); core2: Zhao et al. (2023); core3: Khalid et al. (2024)

Table C.6: ANOVA (160 POI)

Source of Variation	**SS**	**df**	**MS**	**F**	**P-value**	**F crit**
Between Groups	2315973970	9	257330441.1	992.685228	0	1.889320933
Within Groups	256634358.5	990	259226.6248			
Total	2572608329	999				

APPENDIX D: BIONOTE

JHAIME JOSE O. CANDO

jhaimecando18@gmail.com

Fifth-year Computer Science student seeking a software development role. Eager to leverage technical skills and create problem-solving abilities to contribute to innovative projects and drive growth in a dynamic team environment.

EXPERIENCE

Sales-Technical Department - OJT - *Print Depot, Inc.*
Jul 11, 2023 - Aug 16, 2023
Assisted the Project Manager in project documentation

EDUCATION

BS Computer Science - *Pamantasan ng Lungsod ng Maynila*
2020 - Present
Member of Google Developer Club PLM (2020-2021)

Senior High School: STEM - *Dominican School Manila*
2018 - 2020
Took Work Immersion at CREOTEC Philippines

PROJECTS

CGrassPLUS - A Programming Language - *Tkinter, Python*
March 2024
Sole developer in a 5-member team

Web-based Faculty Load and Class Scheduling System for PLM - Computer Science Department - *ExpressJS, EJS, MySQL*
Jun 2022 - Oct 2022
Contributed as on of the Back-end Developers
Created and managed the relational database schema

CERTIFICATION

CS50x: Introduction to Computer Science - *Harvard University (Short Course - Online)*
Jul 2021 - Dec 2022

SKILLS

Technologies

Flask

Django

Tkinter

ExpressJS

Xampp

GIT

Linux

Additional Skills

Problem Solving

Strong attention to detail

Ability to work independently and collaboratively

Eager to learn

SHELLEY PE MANAOIS

2437 Nakar Street, Manila, PH
(+63) 968-763-0514
smanaois25@gmail.com
https://www.linkedin.com/in/mshelleype

Hardworking and passionate individual aiming to use my knowledge and skills to work in an environment which encourages me to succeed and grow professionally

EXPERIENCE

Church Volunteer: Multimedia/Tech - *Quezon Manila*
2022 - Present
Film and video editor, PowerPoint, mini-movies, animation loops

Home Based Job - *Compass Star Ltd.*
2022 - Present

EDUCATION

BS Computer Science - *Pamantasan ng Lungsod ng Maynila*
2020 - Present
Member of Google Developer Club PLM
Member of Computer Science Society
Coursework in UI/UX
Dean’s list

Senior High School: ABM- *Jesus Reigns Christians Academy Manila*
2018 - 2020
Awarded literature outstanding award
Consistent with Honors 2018-2020

ACCOMPLISHMENT

Philippine Android Weekend 2020
UI/UX Beginners - Great Learning
def IT pantry (): A webinar series for the latest web tech 2021
Certified Microsoft Innovative Educator 2020

SKILLS

Technologies

WordPress

Figma

HTML & CSS

Microsoft Office

Additional Skills

Creativity

Verbal Skills

Communication Skills

ANGELO VERANO

Blk 4 Lot 40 Model Community, Tondo, Manila
(+63) 976-055-9167
veranoangelo1530@gmail.com

Currently studying as a 5th year student in Pamantasan ng Lungsod ng Maynila. To be part of an institution that will help me improve, gain new knowledge and skills. Lastly, to help me to become better in the work field that I can use in my future career.

EXPERIENCE

Book Sales Analyst - *Rex Book Store, Inc.*
2020
Manage the book sales monthly and yearly from different schools around Metro Manila and the income they've had through the past months and years

Support Programmer - *Himlayang Pilipino Cemetery*
2022 - 2023
Create a Navigation System of the cemetery thru GIS so that visitors can easily track and find the tombs that they're finding.

GIS Automation Section and Data Processing Section - *Manila Innovation Integrated GIS and Data Hub*
2024 - Present
Managing Data of Manila for improvement and making a comprehensive dashboard for meetings, exhibits and LGU benchmarkings

EDUCATION

BS Computer Science - *Pamantasan ng Lungsod ng Maynila*
2020 - Present

Senior High School - *Holy Child Catholic School*
2018 - 2020

SKILLS

Project Management

Problem Solving

Creativity

Time Management

APPENDIX E: CERTIFICATE OF PRESENTATION

COMPUTER SCIENCE DEPARTMENT
Pamantasan ng Lungsod ng Maynila

CERTIFICATE

OF PRESENTATION

This certificate is presented to

JHAIME JOSE O. CANDO
SHELLEY PE L. MANAOIS
ANGELO P. VERANO

For having presented the research paper titled **Enhancement of Tabu Search Algorithm for Optimized Rescue Routing Operation** during the Computer Science Colloquium 2025 entitled **To CS To Believe: Building the Future, One Bit at a Time** held on May 5-6, 2025 .

Given this 6th day of May 2025 at Bukod Tanging Bulwagan, Pamantasan ng Lungsod ng Maynila

Prof. Raymund M. Dioses, MAED
Chairperson, Computer Science Department

APPENDIX F: ORIGINALITY REPORT

ENHANCEMENT OF TABU SEARCH ALGORITHM FOR OPTIMIZED RESCUE ROUTING OPERATION

ORIGINALITY REPORT

14%	11%	10%	9%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Pamantasan ng Lungsod ng Maynila Student Paper	1%
2	www.mdpi.com Internet Source	1%
3	ouci.dntb.gov.ua Internet Source	1%
4	www.coursehero.com Internet Source	1%
5	Submitted to University of Sunderland Student Paper	<1%
6	revistas.usal.es Internet Source	<1%
7	roadem2024.sciencesconf.org Internet Source	<1%
8	"Handbook of Formal Optimization", Springer Science and Business Media LLC, 2024 Publication	<1%
9	eprints.utm.my Internet Source	<1%
10	manualzz.com Internet Source	<1%

11	www.journalmodernpm.com Internet Source	<1 %
12	www.researchgate.net Internet Source	<1 %
13	Submitted to Colorado Technical University Online Student Paper	<1 %
14	online-journals.org Internet Source	<1 %
15	Alemão, Duarte José Marques. "Dynamic Scheduling for Maintenance Tasks Allocation Supported by Genetic Algorithms", Universidade NOVA de Lisboa (Portugal), 2024 Publication	<1 %
16	ijfmr.com Internet Source	<1 %
17	Submitted to University of Portsmouth Student Paper	<1 %
18	Shixing Zheng. "Improvement strategies for heuristic algorithms based on machine learning and information concepts: a review of the seahorse optimization algorithm", PeerJ Computer Science, 2025 Publication	<1 %
19	etd.aau.edu.et Internet Source	<1 %
20	Submitted to University of Huddersfield Student Paper	<1 %

fdocuments.net

21	Internet Source	<1 %
22	Submitted to De La Salle University - Manila Student Paper	<1 %
23	m.scirp.org Internet Source	<1 %
24	Biao Yuan, Yuwei Lu, Kangzhou Wang, Xiaoshuai Peng. "Multi-objective optimization of the auto-carrier loading and routing problem in an automotive logistics company", Journal of the Operational Research Society, 2023 Publication	<1 %
25	Submitted to University of Cambridge Student Paper	<1 %
26	Submitted to Van Hal Larenstein (VHL) Student Paper	<1 %
27	file.scirp.org Internet Source	<1 %
28	Submitted to American College of the Middle East Student Paper	<1 %
29	Submitted to Georgia Institute of Technology Main Campus Student Paper	<1 %
30	Submitted to University of Teesside Student Paper	<1 %
31	prl-theworkshop.github.io Internet Source	<1 %

32	www.teckelkvalley.com Internet Source	<1 %
33	Submitted to University College London Student Paper	<1 %
34	www.peeref.com Internet Source	<1 %
35	boa.unimib.it Internet Source	<1 %
36	psasir.upm.edu.my Internet Source	<1 %
37	www.jonuns.com Internet Source	<1 %
38	Erlianasha Samsuria, Mohd Saiful Azimi Mahmud, Norhaliza Abdul Wahab, Muhammad Zakiyullah Romdlony et al. "An improved adaptive fuzzy-genetic algorithm based on local search for integrated production and mobile robot scheduling in job-shop flexible manufacturing system", Computers & Industrial Engineering, 2025 Publication	<1 %
39	Kiyofumi Katayama, Kazuo Tanaka, Masahiro Tanaka. "Accurate numerical analysis of resonances in random waveguides: Effects of the waveguide length", AIP Advances, 2024 Publication	<1 %
40	zenodo.org Internet Source	<1 %
41	digibuo.uniovi.es	

	Internet Source	<1%
42	pdfcoffee.com Internet Source	<1%
43	Submitted to Aston University Student Paper	<1%
44	Submitted to Blue Mountain Hotel School Student Paper	<1%
45	etd.repository.ugm.ac.id Internet Source	<1%
46	Submitted to Naval Education Training and Doctrine Command Student Paper	<1%
47	Submitted to University of Durham Student Paper	<1%
48	Submitted to University of Economics, Prague Student Paper	<1%
49	Xiaochen Chou, Luca Maria Gambardella, Roberto Montemanni. "A Tabu Search algorithm for the Probabilistic Orienteering Problem", Computers & Operations Research, 2020 Publication	<1%
50	digilib.library.usp.ac.fj Internet Source	<1%
51	doaj.org Internet Source	<1%
52	etd.hu.edu.et Internet Source	<1%

53	Submitted to Polytechnic University of the Philippines - Sta. Mesa Student Paper	<1 %
54	Submitted to Universiti Tenaga Nasional Student Paper	<1 %
55	Submitted to University of Hong Kong Student Paper	<1 %
56	assets-eu.researchsquare.com Internet Source	<1 %
57	link.springer.com Internet Source	<1 %
58	ideas.repec.org Internet Source	<1 %
59	Ran Tai, Dantong Ouyang, Ruizhi Li, Liming Zhang. "ILSGVCP: An improved local search algorithm for generalized vertex cover problem", Journal of the Operational Research Society, 2023 Publication	<1 %
60	Ruizhi Li, Shuli Hu, Yiyuan Wang, Minghao Yin. "A local search algorithm with tabu strategy and perturbation mechanism for generalized vertex cover problem", Neural Computing and Applications, 2016 Publication	<1 %
61	addi.ehu.es Internet Source	<1 %
62	www2.mmu.ac.uk Internet Source	<1 %

63	C.S.G. Lee, P.R. Chang. "Efficient parallel algorithms for robot forward dynamics computation", IEEE Transactions on Systems, Man, and Cybernetics, 1988 Publication	<1 %
64	dias.library.tuc.gr Internet Source	<1 %
65	www.biopharmaservices.com Internet Source	<1 %
66	Arnaldo Pérez Castaño. "Practical Artificial Intelligence", Springer Science and Business Media LLC, 2018 Publication	<1 %
67	Submitted to Monash University Student Paper	<1 %
68	Saeed Saeedvand, Hadi S. Aghdasi, Jacky Baltes. "Novel hybrid algorithm for Team Orienteering Problem with Time Windows for rescue applications", Applied Soft Computing, 2020 Publication	<1 %
69	Safaa Safouan, Karim El Moutaouakil. "Enhanced fractional probabilistic self-organizing maps with genetic algorithm optimization (EF-PRSOM)", Evolutionary Intelligence, 2025 Publication	<1 %
70	www.bu.edu Internet Source	<1 %

Hanoi Pedagogical University 2

71	Publication	<1 %
72	Saiqi Zhou, Dezhi Zhang, Bin Ji, Shaoyu Zhou, Shuangyan Li, Likun Zhou. "A MILP model and heuristic method for the time-dependent electric vehicle routing and scheduling problem with time windows", Journal of Cleaner Production, 2024 Publication	<1 %
73	openrepository.aut.ac.nz Internet Source	<1 %
74	paper.ijcsns.org Internet Source	<1 %
75	press.uhnsugriwa.ac.id Internet Source	<1 %
76	worldwidescience.org Internet Source	<1 %
77	www.db-thueringen.de Internet Source	<1 %
78	www.gambetta.org Internet Source	<1 %
79	www.rairo-ro.org Internet Source	<1 %
80	www.sciencedirect.com Internet Source	<1 %
81	Matias Greco, Jorge Toro, Carlos Hernández, Jorge A. Baier. "K Focal Search for Slow Learned Heuristics", IEEE Access, 2023 Publication	<1 %

82	"Innovations in Machine and Deep Learning", Springer Science and Business Media LLC, 2023 Publication	<1 %
83	Cláudio Félix Canguende-Valentim, António Carrizo Moreira, Vera Teixeira Vale. "Counterfeit Luxury and Consumption", Routledge, 2025 Publication	<1 %
84	YiWen Zhong. "The Study of Neighborhood Structure of Tabu Search Algorithm for Traveling Salesman Problem", 2008 Fourth International Conference on Natural Computation, 10/2008 Publication	<1 %
85	animorepository.dlsu.edu.ph Internet Source	<1 %

Exclude quotes	On	Exclude matches	< 5 words
Exclude bibliography	On		