The Pollution Traveling Salesman Problem with Refueling

Panagiotis Karakostas, Angelo Sifaleras[*]

Department of Applied Informatics, School of Information Sciences, University of Macedonia,
156 Egnatias Str., Thessaloniki 54636, Greece

Abstract

This study presents the Pollution Traveling Salesman Problem with Refueling, a novel optimization problem which integrates two recently proposed variants of the Traveling Salesman Problem: the Pollution Traveling Salesman Problem and the Traveling Salesman Problem with Refueling. The proposed problem captures the operational dynamics of a real-world routing scenario involving a single vehicle originating from a central depot and delivering products to end customers. When considering the vehicle's fuel tank capacity and fuel consumption during the routing process, the need to visit fuel stations for refueling arises. To address this complex problem, a new mixed integer linear programming model was developed, and the Gurobi solver was employed to solve smaller instances. For the effective resolution of larger practical problem cases, a two-stage double adaptive general variable neighborhood search method was proposed. The proposed methodology exhibits comparable efficiency to a commercial solver, demonstrating notably low execution time requirements. To further assess its performance, a comparative study was conducted on TSPLib instances. In comparison to various solution approaches documented in the open literature, encompassing both VNS-based and alternative methods, our proposed approach consistently yields highly competitive results within low execution times.

Keywords: Metaheuristics, Variable Neighborhood Search, Adaptive Search, Intelligent Optimization, Traveling Salesman Problem

[*] Corresponding author

Email addresses: pankarakostas@uom.edu.gr (Panagiotis Karakostas), sifalera@uom.gr (Angelo Sifaleras)

1. Introduction

Distribution of end products constitutes a critical component of outbound shipping logistics operations (Archetti et al., 2022). These operations require the optimal configuration of different decisions, such as fleet-scheduling and routing decisions (Speranza, 2018). In the case of small-sized, and more rarely medium-sized, enterprises, such processes are performed by a single vehicle, which should serve a predefined number of customers in a single time period. However, the Traveling Salesman Problem (TSP), while offering an abstract representation of such operations, fails to account for crucial real-world aspects (Kowalik et al., 2023), such as fuel consumption, CO_2 emissions, and refueling requirements (Neves-Moreira et al., 2020; Cacchiani et al., 2023). To this end, this work introduces a novel TSP variant, the Pollution TSP with Refueling (PTSPR), which extends the classic TSP by considering fuel consumption, taxation over the emitted pollutants, driver wages, and refueling decisions. The introduced optimization problem offers avenues for addressing various practical scenarios that entail the utilization of a single conventional vehicle. These applications encompass domains such as home heating oil delivery, the distribution of bakery products, the provision of home healthcare services, regional surveillance operations, and agricultural pesticide applications. The PTSPR is a combinatorial optimization problem that integrates two newly proposed TSP variants, the Pollution TSP (PTSP) and the TSP with Refueling (TSPWR). To this end, the present work addresses the following research contributions:

- We introduce the PTSPR as a realistic extension of the classic TSP, which considers the well-known Comprehensive Modal Emission Model (CMEM), as well as routing and refueling decisions.

- A mixed-integer linear programming (MILP) model is proposed to mathematically formulate the new combinatorial optimization problem.

- A two-stage Double Adaptive General Variable Neighborhood Search (DA-GVNS)-based solution method is proposed for the effective solution of large problem instances.

- An extended computational analysis is presented to justify the proper tuning of the proposed algorithm.

- Sensitivity analyses are conducted to investigate the potential impact of fuel tank size and refueling policy fluctuations on the network structure and its total cost.

This work is structured as follows. Section 2 provides a literature review of the related research works. The problem statement and its mathematical formulation are provided in Section 3. Section 4 presents the proposed solution method, while the computational results are given in Section 5. Finally, Section 7 summarizes the concluding remarks and potential future extensions.

2. Literature review

The PTSP (Cacchiani et al., 2018, 2023) extends the asymmetric TSP by considering fuel consumption and CO_2 emissions using the well-established CMEM of Barth et al. (2005; 2009), as well as the cost of driver wages. Specifically, the PTSP involves a solitary capacitated vehicle embarking from a designated depot to serve a set of geographically dispersed customers, each characterized by fixed nonnegative demands and service times. The primary objective of the PTSP is to determine an optimal route that minimizes both the total fuel consumption, measured in liters, and the driver wages cost, expressed in British pounds. A distinctive feature of this problem variant lies in the incorporation of the CMEM, which systematically calculates the consumed fuel of the vehicle. This calculation takes into account various factors, including the traveled distance, vehicle speed, and the total weight of the vehicle, encompassing both curb weight and load.

The second problem, the TSPWR (Ottoni et al., 2022) is based on the Gas Station Problem (Khuller et al., 2007) and extends the classic TSP by considering refueling decisions. More specifically, the authors examine a scenario involving a mobile agent departing from a depot, tasked with visiting a set of cities and determining optimal refueling locations before returning to the depot. The primary objective is to minimize the total route cost, with a focus on refueling expenses. Building upon the Gas Station Problem presented by

Khuller et al. (2007), the study encompasses two distinct problem cases. In the first case, fixed fuel prices are taken into account, while the second case introduces variations in fuel prices. Furthermore, the TSPWR accommodates the possibility of necessitating refueling along a road link. In such instances, the consideration of a tow truck is incorporated, introducing an additional, arc-based cost.

Several research contributions in the open literature have considered the integration of analytical fuel consumption models with routing decisions (Bektaş & Laporte, 2011; Koç et al., 2014; Cheng et al., 2017; Karakostas et al., 2020, 2022). However, these research approaches have not focused on the consideration of refueling decisions. Although several research works have addressed refueling decisions in routing-based optimization problems (Suzuki, 2012, 2014; Goeke & Schneider, 2015; Schiffer et al., 2018; Neves-Moreira et al., 2020), they have mainly addressed problem variants with fixed routes or edge-based fuel consumption weights/rates. To the best of our knowledge, there is a research gap in the integration of routing-based optimization problems with analytical fuel consumption models and refueling decisions. Incorporating analytical fuel consumption models and refueling decisions in routing-based optimization problems offers several benefits (Dukkanci et al., 2019). By accurately estimating fuel usage, routes can be optimized to enhance efficiency and reduce costs. Analyzing carbon emissions allows for environmentally conscious routing decisions. Optimal refueling strategies lead to significant cost savings (Neves-Moreira et al., 2020). In addition, incorporating these models adds realism to transportation system simulations, aiding in reliable performance assessments. Therefore, analytical fuel consumption models and refueling decisions play a crucial role in advancing transportation research and decision-making processes.

3. Problem statement and mathematical formulation

The PTSPR constitutes a novel combinatorial optimization problem designed to incorporate additional realistic features into the optimal routing of a single vehicle. This problem emerges from the conceptual integration of the PTSP and the TSPR. Specifically, PTSPR adopts the valid CMEM from PTSP to comprehensively calculate both the consumed fuel and emitted CO_2 of the vehicle. Drawing inspiration from TSPR, it incorporates the notion of refueling. However, PTSPR significantly distinguishes itself from both PTSP and TSPR. Notably, PTSPR extends PTSP by introducing refueling decisions within the CMEM, accounting for varying fuel prices. Additionally, PTSP integrates units representing distinct quantities, such as liters of fuel and monetary cost, into a unified objective function, a practice not commonly observed in optimization. In our methodology, fuel consumption is translated into monetary units, considering fuel prices and taxation costs per unit of emitted pollutants. Another departure from PTSP involves the exclusion of fixed service times. This exclusion is motivated by the realization that fixed service times, remaining constant regardless of node visitation sequence, contribute a constant value to the objective function, prompting their exclusion for simplicity. In contrast to TSPR, PTSPR introduces several distinctions. It incorporates an analytical model to calculate vehicle fuel consumption and introduces a unique set of nodes representing gas stations with varying fuel prices. This is a departure from TSPR's assumption that refueling can occur at any main node of the problem. Furthermore, it is essential to clarify that, in this initial approach of the PTSPR, considerations related to speed limits are omitted. Hence, the addressed problem is more suitably configured to accommodate inter-city delivery operations.

The PTSPR is defined on a complete graph $G = \{N, E\}$, which consists of the set of nodes N, which is partitioned into the set of operational vertices (depot, its duplicate and n geographically dispersed customers) $I = \{0, ..., (n+1)\}$ and the set of fuel station vertices $J = \{n+2, ..., n+|J|\}$ and the set of edges $E = \{(i,j) : i, j \in N, \ i \neq j\}$. Each edge $(i,j) \in E$ is associated with a distance $c_{i,j}$. Node 0 and its duplicate, $(n+1)$, denote the depot of the logistics system under consideration. Each customer has a positive demand, $d_i, i \in I - \{0,(n+1)\}$, which should be fully satisfied in a single time period by a single vehicle with capacity $Q = \sum_{i \in I - \{0,(n+1)\}} d_i$. The vehicle can traverse an active link of the network with a specific speed level, $l \in L = \{1, ..., |L|\}$.

The vehicle departs from the depot 0 fully loaded and fueled, considering an upper fuel limit (UFL). The fuel level gradually decreases as the vehicle passes a selected edge

$(i,j) \in E$ according to the fuel consumption that occurred. Fuel consumption is calculated in proportion to the load and speed of the vehicle, as well as the distance that should be covered. Whenever the fuel level of the vehicle is less than or equal to a specific limit (Lowest Fuel Limit - LFL), it should visit a fuel station $j \in J$ to refuel. Each fuel station $j \in J$ has a unitary fuel price, FP_j.

In the context of the PTSPR, this problem entails several critical components and objectives. These encompass the knowledge of the geographical location of the depot, a set of geographically dispersed customers, each characterized by a fixed demand, a collection of available fuel stations offering varying fuel prices, specific vehicle attributes, and a selection of available speed levels. The principal objective of PTSPR is to determine the optimal sequence for servicing customers, select the most suitable speed levels for traveling between network nodes, identify efficient refueling locations, and calculate the requisite fuel quantities. This optimization seeks to minimize the comprehensive cost structure, consisting of driver wage expenses, refueling outlays, and emissions taxation charges. In formulating the PTSPR, several assumptions are employed to streamline and clarify the problem. These assumptions include the fulfillment of all customer demands within a single time period, the initiation of the vehicle journey from the depot and its return to the same location, permitting multiple visits to fuel stations, and the absence of capacity constraints for the fuel stations.

The PTSPR has been mathematically formulated as an MILP model. Tables 1, 2, 3, and 4 summarize the sets, the parameters (vehicle-based and general), and the decision variables of the model.

Table 1: Model sets

Set	**Explanation**
N	Set of nodes
I	Set of operational nodes (depot and customers)
IO	Set of customers
J	Set of fuel stations
NNO	Set of all nodes excluded the depot ($NNO = N - \{0\}$)
NNL	Set of all nodes excluded the duplicate of the depot ($NNL = N - \{(n+1)\}$)
L	Set of the available speed levels

Table 2: Vehicle-based parameters

Parameter	**Explanation**	**Value** **(Cheng et al.,2017)**
ϵ	fuel-to-air mass ratio	1
g	gravitational constant (m/s^2)	9.81
ρ	air density (kg/m^3)	1.2041
CR	coefficient of rolling resistance	0.01
η	efficiency parameter for diesel engines	0.45
$HVDF$	heating value of a typical diesel fuel (kj/g)	44
ψ	conversion factor (g/s to L/s)	737
θ	road angle	0
τ	acceleration (m/s^2)	0
CW	curb weight (kg)	4672
EFF	engine friction factor ($kj/rev/L$)	0.25
ES	engine speed (rev/s)	39
ED	engine displacement (L)	2.77
CAD	coefficient of aerodynamics drag	0.6
FSA	frontal surface area (m^2)	9
$VDTE$	vehicle drive train efficiency	0.4
Q	capacity of the vehicle	*Instance-dependent*

Table 3: Remaining model parameters

Parameter	Explanation	Value
ET	unit CO_2 emission cost ($Euros/kg$)	0.29
DW	driver wage ($Euros/s$)	0.0025
em	CO_2 emitted by unit fuel consumption	2.699
$c_{i,j}$	distance between nodes i and j in meters	*Instance-dependent*
d_i	demand of customer i	*Instance-dependent*
FP_j	unitary fuel price at fuel station j	*Instance-dependent*
UFL	upper fuel limit (fuel tank size) in litre	85
LFL	lowest fuel limit	$0.25 \cdot UFL$
s_l	Speed level $l \in L$ (m/s)	$(8.33, 11.11, 13.89, 16.67, 19.44,$ $22.22, 25, 27.78, 29.17, 30.56)$

Table 4: Model variables

Decision Variable	Explanation
x_{ij}	binary variable which denotes if the vehicle moves from node $i \in N$ to node $j \in N$ or not
z_{ijl}	binary variable which denotes if the vehicle moves from node $i \in N$ to node $j \in N$ with a speed level $l \in L$ or not
w_i	binary variable which denotes if the node $i \in N$ is part of the route or not
f_{ij}	positive continuous variable which denotes the load of vehicle when moving from node $i \in N$ to node $j \in N$
r_{ij}	positive continuous variable which denotes the refueled quantity at node $j \in N$ after the service of node $i \in N$
FL_{ij}	positive continuous variable which denotes the fuel level of vehicle when arrives at node $j \in N$ after the service of node $i \in N$
$FCons_{ij}$	positive continuous variable which denotes the fuel quantity consumed moving from node $i \in N$ to node $j \in N$

Here, we provide the necessary mathematical expressions to simplify the function which calculates the fuel consumption following the CMEM. More specifically, $\lambda = \frac{\epsilon}{HVDF \cdot \psi}$, $\gamma = \frac{1}{1000 \cdot VDTE \cdot \eta}$, $\alpha = \tau + g \cdot CR \cdot \sin\theta + g \cdot CR \cdot \cos\theta$, and $\beta = 0.5 \cdot CAD \cdot \rho \cdot FSA_k$.

The proposed MILP model of the PTSPR is formulated as follows:

$$\begin{aligned} \min \sum_{i \in N} \sum_{j \in N} & ET \cdot em \cdot FCons_{ij} \\ + \sum_{i \in I} & r_{ij} \cdot \sum_{j \in J} FP_j \\ + \sum_{i \in N} \sum_{j \in N} \sum_{l \in L} & z_{ijl} \cdot \tfrac{c_{ij}}{s_l} \cdot DW \end{aligned} \tag{1}$$

Subject to

$$\sum_{i \in I} x_{ij} \geq w_j, \quad \forall j \in J \tag{2}$$

$$w_i = 1, \quad \forall i \in IO, \text{ if } d_i > 0 \tag{3}$$

$$x_{ij} \leq \frac{1}{2} \cdot (w_i + w_j), \; \forall i \in NNL, \; \forall j \in NNO \tag{4}$$

$$\sum_{j \in NNO} x_{0j} = 1 \tag{5}$$

$$\sum_{j \in NNL} x_{j(n+1)} = 1 \tag{6}$$

$$\sum_{j \in NNO} x_{ij} = w_i, \quad \forall i \in I, i \neq (n+1) \tag{7}$$

$$\sum_{j \in NNL} x_{ji} = w_i, \quad \forall i \in I, i \neq 0 \tag{8}$$

$$\sum_{i \in NNL} x_{ij} - \sum_{i \in NNO} x_{ji} = 0, \; \forall j \in N - \{0, (n+1)\} \tag{9}$$

$$x_{ij} = 0,\ \forall i, j \in J \tag{10}$$

$$\sum_{l \in L} z_{ijl} = x_{ij},\ \forall i \in N, i \neq j,\ \forall j \in N \tag{11}$$

$$\sum_{j \in N - \{0, n+1\}} f_{0j} = Q \tag{12}$$

$$\sum_{j \in N - \{0, n+1\}} f_{j(n+1)} = 0 \tag{13}$$

$$\sum_{j \in N, j \neq i} f_{ji} - \sum_{j \in N, j \neq i} f_{ij} = d_i,\ \forall i \in N - \{0, (n+1)\} \tag{14}$$

$$f_{ij} - (Q \cdot (1 - x_{ij})) \leq \sum_{u \in I} f_{ui} - d_i,\ \forall i \in I,\ \forall j \in J \tag{15}$$

$$f_{ji} - (Q \cdot (1 - x_{ji})) \leq \sum_{u \in I} f_{uj},\ \forall i \in I,\ \forall j \in J \tag{16}$$

$$f_{ij} \leq (Q - d_i) \cdot x_{ij},\ \forall i \in N,\ \forall j \in N \tag{17}$$

$$FL_{ui} + r_{ui} + (UFL \cdot (1 - x_{ui})) - FCons_{ij} \geq FL_{ij} - (UFL \cdot (1 - x_{ij})),\ \forall u \in N,\ \forall i \in N,\ \forall j \in N \tag{18}$$

$$FL_{ui} + r_{ui} - (UFL \cdot (1 - x_{ui})) - FCons_{ij} \leq FL_{ij} + (UFL \cdot (1 - x_{ij})),\ \forall u \in N,\ \forall i \in N,\ \forall j \in N \tag{19}$$

$$FL_{0i} \leq UFL - FCons_{0i} + (UFL \cdot (1 - x_{0i})),\ \forall i \in NNO \tag{20}$$

$$FL_{0i} \geq UFL - FCons_{0i} + (2 \cdot UFL \cdot (1 - x_{0i})),\ \forall i \in NNO \tag{21}$$

$$FL_{ij} \leq UFL \cdot x_{ij},\ \forall i \in NNL,\ \forall j \in NNO \tag{22}$$

$$FL_{ij} \geq LFL \cdot x_{ij},\ \forall i \in NNL,\ \forall j \in NNO \tag{23}$$

$$r_{ij} \leq x_{ij} \cdot (UFL - LFL),\ \forall i \in NNL,\ \forall j \in J \tag{24}$$

$$r_{ij} + (UFL \cdot (1 - x_{ij})) \geq UFL - (FL_{ui} - FCons_{ij}) - (UFL \cdot (1 - x_{ui})),\ \forall i \in NNO,\ \forall u \in NNL,\ \forall j \in J \tag{25}$$

$$\sum_{i \in N} r_{ij} = 0,\ \forall j \in I \tag{26}$$

$$FCons_{ij} \geq \lambda \cdot \{ (\sum_{l \in L} \frac{z_{ijl} \cdot EFF \cdot ES \cdot ED \cdot c_{ij}}{s_l}) + \big(\alpha \cdot \gamma \cdot ((CW \cdot x_{ij}) + f_{ij}) \cdot c_{ij} \big) + \big(\beta \cdot \gamma \cdot \sum_{l \in L} z_{ijl} \cdot s_l^2 \big) \},\ \forall i \in N,\ \forall j \in N \tag{27}$$

$$x_{ij} \in \{0, 1\},\ \forall i, j \in N, i \neq j \tag{28}$$

$$z_{ijl} \in \{0, 1\},\ \forall i, j \in N, i \neq j,\ \forall l \in L \tag{29}$$

$$w_i \in \{0, 1\},\ \forall i \in N \tag{30}$$

$$f_{ij} \geq 0,\ \forall i, j \in N \tag{31}$$

$$FL_{ij},\ r_{ij},\ FCons_{ij} \in [0, UFL],\ \forall i, j \in N \tag{32}$$

The first component of the objective function of the PTSPR (1) represents the taxation cost of the pollutants emitted due to fuel consumption. The next component denotes the refueling cost, while the last provides the wage cost of the driver. Constraints (2) guarantee that if a fuel station is selected to be part of the route, it should be visited at least once. All customers of the logistics system must be included in the constructed route, as indicated by constraints (3). Constraints (4) guarantee that the vehicle will move between two nodes only if they are selected to be part of the route. Constraints (5) ensure the departure of the vehicle from the depot, while constraints (6) impose its return to the depot. Constraints (7) and (8) guarantee that the vehicle will serve each customer once. Similarly, constraints (9) impose the balance between the incoming and outgoing flows of the vehicle for each node included in the constructed route. The vehicle cannot move from one fuel station to another, as imposed by constraints (10). Vehicle movement at each active edge of the constructed route must be carried out with a specific speed level imposed by constraints (11). Depending on the load of the vehicle, constraints (12) ensure that the vehicle departs from the depot fully loaded, while constraints (13) guarantee that it returns to the depot empty. Equations (14), (15) and (16) constitute load balance constraints. More specifically, constraints (14) refer to vehicle movements between customer nodes, while constraints (15) and (16) refer to the load balance in the event of movement from customer to fuel station and from a fuel station to a customer, respectively. Constraints (17) set an upper bound for the load of the vehicle on each selected route. Constraints (18), (19), (20), and (21) impose the balance of the fuel levels at each node of the structured logistics system taking into account the possible refueling processes and fuel consumption. To clarify, constraints (20) and (21) refer to the balance of the fuel level in the fuel tank of the vehicle while it moves from the depot to any other node of the system. Load and fuel balance constraints act as subtour elimination constraints. Constraints (22) and (23) establish upper and lower bounds of fuel level in the vehicle fuel tank. Similarly, the constraints (24) and (25) set upper and lower bounds on the quantity of refueling in the fuel stations visited. Constraints (26) impose that no refueling will be performed at the customer locations. Constraints (27) refer to the calculation of the fuel consumption for each active link in the constructed route. Constraints (28)-(32)

represent the domain of the decision variables.

4. Solution approach

The PTSPR is a complex optimization problem. Furthermore, considering load and fuel level balance constraints significantly increases its complexity. To this end, the utilization of exact algorithms through commercial optimization solvers can contribute to the solution of only small-scale problem instances. Thus, we focused on the development of a heuristic algorithm for the solution of realistic size problem instances.

More specifically, a two-stage heuristic solution approach was designed considering the fact that on an optimized route, the additional stops required for refueling are eliminated. Furthermore, taking advantage of recent findings from Karakostas & Sifaleras (2022), we developed a two-stage DA-GVNS-based heuristic method. The first stage is dedicated to the optimization of the route, while the second focuses on fuel-based decisions. The overall algorithmic approach is illustrated in Figure 1.

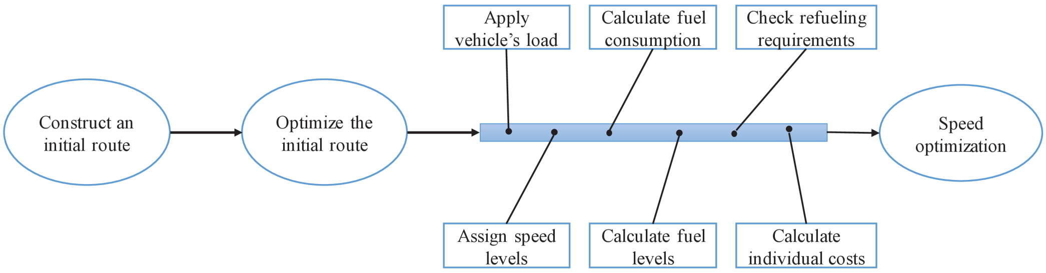

Construct an initial route
Optimize the initial route
Apply vehicle's load
Calculate fuel consumption
Check refueling requirements
Assign speed levels
Calculate fuel levels
Calculate individual costs
Speed optimization

Figure 1: Proposed solution approach

The Nearest-Neighbor Heuristic (NNH) (Flood, 1956) is used to build an initial feasible route, and a DA-GVNS is applied to optimize the initial route. Next, all the processes depicted in the blue bar in Figure 1 are performed. If this process leads to refueling requirements, the next steps will be applied. Otherwise, the solution method is completed by applying a speed optimization method.

In the case of refueling requirements, a modified iterative heuristic method, based on the Cheapest Insertion (Johnson & Papadimitriou, 1985), is applied to address refueling decisions. Next, the speed optimization method is applied.

Algorithm 1 provides the overall proposed solution method for the PTSPR. At the end of each line, a link is provided to the appropriate section as a step-by-step explanation of the pseudo-code. The overall solution approach typically begins by computing an initial feasible route using the NNH (line 3). Subsequently, this route undergoes further enhancement through a DA-GVNS method (line 4). Following this, for each link in this cost-effective route, the load of the vehicle is assigned based on the demands of each customer (line 5), and the fuel consumption, as well as the fuel levels at each node, are calculated (lines six and seven, respectively). The eighth line of the pseudo-code denotes the application of a check routine to assess if refueling is necessary at any point along the formed route. Should refueling be required, fuel station nodes are selected using a modified CI approach (line 10), and the process concludes by optimizing the speed levels of the vehicle for each link of the route (line 11). Alternatively, if refueling is not necessary, only the speed optimization procedure is executed (line 14).

Algorithm 1 Proposed solution method for the PTSPR

```
procedure $TwoStage\_DA-GVNS(S, l2_{max}, FailuresLimit, Local\_TimeLimit)$
    $Assign\ values\ to\ the\ arguments\ of\ DA-GVNS$    ▷ Section 4.2.6
    $S' \leftarrow NNH$    ▷ Section 4.2.1
    $S \leftarrow DA-GVNS(S', k_{max}, max\_time, l_{max}, Sh_{max}, Initial\_LS\_Order, Initial\_Shaking\_Order)$    ▷ Section 4.2.2
    $S' \leftarrow ApplyLoads(S)$    ▷ Section 4.3.1
    $S \leftarrow CalculateFuelConsumption(S')$    ▷ Section 4.3.3
    $S' \leftarrow CalculateFuelLevels(S)$    ▷ Section 4.3.4
    $Check\ Refueling\ Requirements$    ▷ Section 4.3.5
    if $RefuelingRequirement = true$ then
        $S \leftarrow modified\_CheapestInsertion(S')$    ▷ Section 4.3.6
        $S' \leftarrow SpeedOptimization(S)$    ▷ Section 4.3.6
        $S \leftarrow S'$
    else
        $S' \leftarrow SpeedOptimization(S)$    ▷ Section 4.3.6
        $S \leftarrow S'$
    end if
end procedure
```

4.1. Solution representation

To provide clarity, a PTSPR solution is denoted as $S = \{RP, RS, SL\}$, where RP represents an array detailing the sequence of customers in the main route of the vehicle, RS takes the form of a matrix that outlines the refueling schedule, and SL is a matrix illustrating the speed levels for traversing the nodes within the integrated route. Specifically, with n representing the count of operational nodes excluding the duplicate depot node, RS is an $n \times n$ matrix with elements in the format $(fuel\_station, refueling\_quantity)$. In instances where no refueling is necessary between operational nodes, the corresponding cell in RS will be denoted as $(0, 0)$. Additionally, SL is a $NNL \times NNL$ matrix, where each cell designates the speed level of the vehicle when traversing the corresponding link. Cells within SL that do not pertain to active route links will be assigned a value of 0. Figures 2 and 3 present an illustrative example of the representation of a PTSPR solution.

$$\overbrace{\text{RP} = [1, 5, 4, 2, 6, 3, 7] \qquad \text{RS} = \begin{array}{c|ccccccc|} & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 1 & (0, 0) & (0, 0) & (0, 0) & (0, 0) & (0, 0) & (0, 0) & (0, 0) \\ 2 & (0, 0) & (0, 0) & (0, 0) & (0, 0) & (0, 0) & (0, 0) & (0, 0) \\ 3 & (0, 0) & (0, 0) & (0, 0) & (0, 0) & (0, 0) & (0, 0) & (1, 40) \\ 4 & (0, 0) & (3, 25) & (0, 0) & (0, 0) & (0, 0) & (0, 0) & (0, 0) \\ 5 & (0, 0) & (0, 0) & (0, 0) & (0, 0) & (0, 0) & (0, 0) & (0, 0) \\ 6 & (0, 0) & (0, 0) & (0, 0) & (0, 0) & (0, 0) & (0, 0) & (0, 0) \\ 7 & (0, 0) & (0, 0) & (0, 0) & (0, 0) & (0, 0) & (0, 0) & (0, 0) \end{array} \qquad \text{SL} = \begin{array}{c|cccccccccc|} & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\ 1 & 0 & 0 & 0 & 0 & 40 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 0 & 0 & 40 & 0 & 0 & 0 & 0 \\ 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 60 & 0 & 0 \\ 4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 50 \\ 5 & 0 & 0 & 0 & 60 & 0 & 0 & 0 & 0 & 0 & 0 \\ 6 & 0 & 0 & 80 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 7 & 50 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 8 & 0 & 0 & 0 & 0 & 0 & 0 & 80 & 0 & 0 & 0 \\ 9 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 10 & 0 & 80 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}}^{\text{S = \{RP, RS, SL\}}}$$

Figure 2: Solution representation example (a)

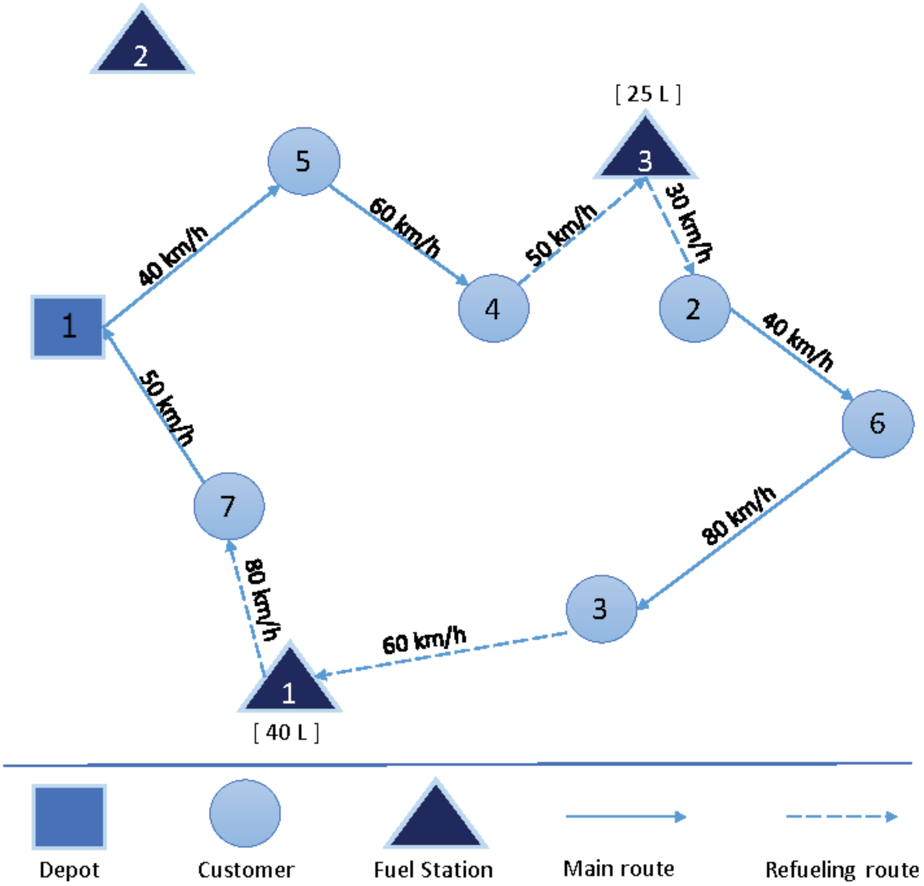


Figure 3: Solution representation example (b)

The provided example comprises one depot (node 1), six customers (nodes 2-7), and three fuel stations. The matrix RP outlines the primary route of the problem, encompassing the depot and the customers. Each cell in the matrix RS represents the refueling plan, indicating the fuel station and the quantity of refueling required. For instance, in Figure 2, the cell at row four and column two signifies that the vehicle, after departing from customer four en route to customer two, should visit fuel station three for a refueling of 25 liters. Similarly, when the vehicle travels from customer three to customer seven, it should visit fuel station one to refuel 40 liters. The combination of matrices RP and RS yields the final integrated route of the problem. Matrix SL encompasses all nodes, including the depot, six customers, and three fuel stations. Therefore, the fuel stations are denoted by the count of the depot and customers plus their corresponding numbers. Consequently, the first fuel station corresponds to the eighth node, the second to the ninth, and the third to the tenth node. In light of these clarifications, matrix SL provides the optimal speed levels for traversing between the edges of the integrated route.

4.2. First stage of the proposed solution method

4.2.1. Construct an initial TSP route

The first step of the proposed solution method focuses on the use of the well-known NNH to build a feasible TSP route starting from the depot, visiting all customers once, and returning to the depot. At this initial step, the metric considered is the distance. From a technical perspective, a vector of length $|I|$ is utilized to store the sequence of nodes.

4.2.2. Optimize the initial TSP route

Variable Neighborhood Search (VNS) is a well-established metaheuristic framework, characterized by its simplicity and efficiency for solving hard optimization problems (Hansen et al., 2017; Brimberg et al., 2023). General VNS (GVNS) constitutes a variant of VNS that has been utilized to efficiently solve several supply chain network optimization problems (Mladenović et al., 2014; Menéndez et al., 2017; Karakostas et al., 2020).

DA-GVNS is a recently proposed GVNS extension, which uses low-level machine learning procedures to reorder the operators used in both the improvement and the shaking phase

(Karakostas & Sifaleras, 2022). The DA-GVNS was proved to be more efficient compared to the conventional GVNS and its single-adaptive variants in the case of TSP. To this end, we developed a DA-GVNS for the optimization of the initial TSP route, which consists of the three following components:

- An adaptive pVND as the main improvement component,
- An adaptive intensified shaking method as the diversification component,
- a solution renewal step.

The structure of the developed DA-GVNS bears a resemblance to the one presented in the work of Karakostas & Sifaleras (2022). However, the proposed DA-GVNS incorporates the following new characteristics:

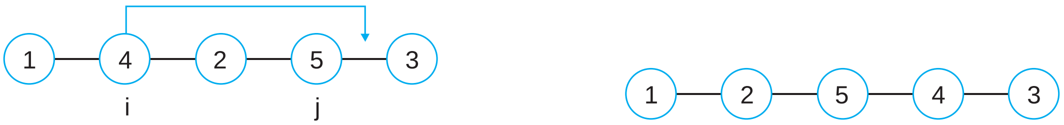
- additional local search operators were examined to determine their suitability for inclusion in the adaptive pVND framework,
- the adaptive search strategy (Karakostas et al., 2019b) was also considered.

4.2.3. Neighborhood structures

The following five neighborhood structures were considered as local search operators in the adaptive pVND developed. These neighborhood structures can be applied under any search strategy, as discussed in Subsection 5.3.

Relocate. This operator selects two nodes and removes the first one, i, from its current location in the route, and re-inserts it next to the second one, j.

Figure 4 illustrates an example of the application of the Relocate operator on a five-node route, where $i = 4$ and $j = 5$.



(a)

(b)

Figure 4: Route instance before (a) and after (b) the application of the Relocate operator

Swap. This operator, also known as *1-1 exchange*, selects two nodes, i and j, from the route and swaps them.

Figure 5 illustrates an example of the application of the swap operator on a five-node route, where $i = 4$ and $j = 5$.

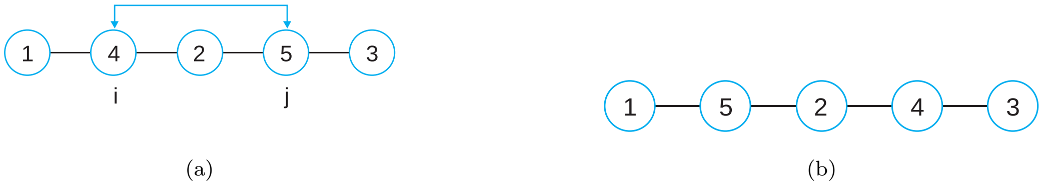


Figure 5: Route instance before (a) and after (b) the application of the swap operator.

2-Opt. This operator breaks two edges $(i, i + 1)$ and $(j, j + 1)$ and reconnects them (Croes, 1958). It can also be implemented as a subtour reversal move (Hillier & Lieberman, 2021).

Figure 6 illustrates an example of the application of the 2-Opt operator on a six-node route, where the first selected edge is $(4, 2)$ and the second is $(3, 6)$.

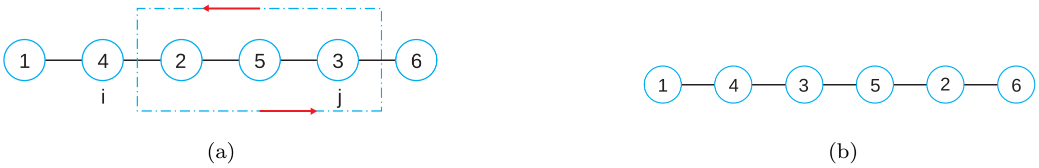


Figure 6: Route instance before (a) and after (b) the application of the 2-Opt operator.

Or-Opt. Typically, this move operates by flexibly relocating one, two, or three consecutive nodes within a solution (Or, 1976). However, in this particular context, its operation has been fixed to the relocation of two consecutive nodes, resembling an Edge Relocation move. Thus, the operator selects an edge $(i, i + 1)$ and a single node j from the route. It removes the edge from its current location and re-inserts it exactly after the node j.

An example of Or-Opt applied on a six-node route, where the selected edge is $(4, 2)$ and $j = 4$, is illustrated in Figure 7.

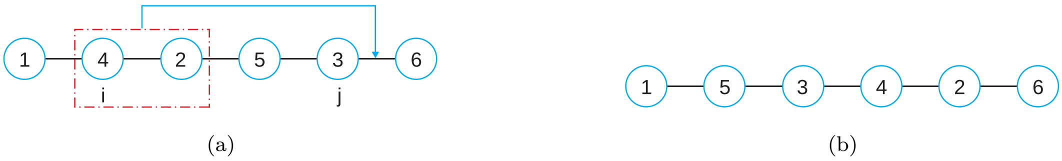
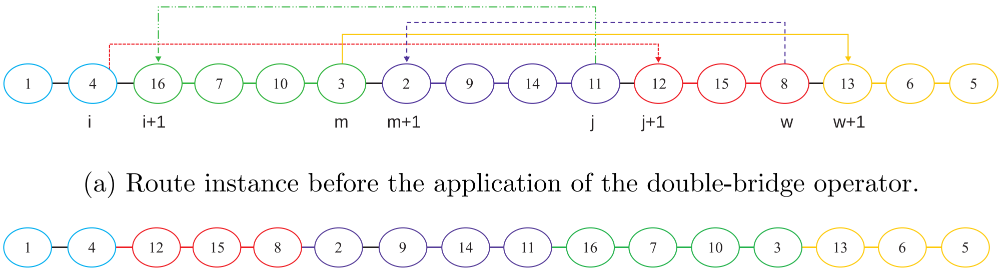


Figure 7: Route instance before (a) and after (b) the application of the Or-Opt operator

Double-Bridge. This operator is a special case of the 4-Opt move in which four edges $\{(i, i+1), (j, j+1), (m, m+1), (w, w+1)\}$ are broken and reconnected differently $\{(i, j+1), (w, m+1), (j, i+1), (m, w+1)\}$. To apply the double bridge, the following rule must be respected: $w > j > m > i$.

Figure 8 provides an illustrative example of the Double-Bridge operator applied on a 16 nodes route. The selected edges, denoted by their positions in the route, are $(2, 3), (10, 11), (6, 7), (13, 14)$. It is obvious that the selected positions respect the application rule, as $w = 13 > j = 10 > m = 6 > i = 2$.



(a) Route instance before the application of the double-bridge operator.

(b) Route instance after the application of the double-bridge operator.

Figure 8: Comparison of route instances before and after the application of the double-bridge operator.

4.2.4. Adaptive pVND

The pVND is an extension of the basic VND method (Hansen et al., 2017), in which each local search operator is applied until no further improvements can be found. Then, the search continues with the next operator. This process continues until all considered local search operators are applied. The adaptive pVND is enriched by an adaptive re-ordering mechanism that redefines the execution order of the considered local search operators, based on the number of improvements they achieved in each previous execution of the method (Todosijević et al., 2016; Karakostas & Sifaleras, 2022). To this end, the adaptive re-ordering mechanism of the local search operators considers:

- The initial order of operators, $Initial\_LS\_Order$, as a vector of length l_{max} (l_{max} denotes the total count of operators used during the improvement phase), which assigns each position to an operator in the first iteration, or
- the current order of operators, $Current\_LS\_Order$, as a vector of length l_{max}, which assigns each position to an operator for the following iterations,
- the number of improvements of each operator stored in the $LS\_Improvements\_Counter$, a vector of length l_{max}. Each position represents an operator and the corresponding value in this position represents the number of improvements achieved using this operator in the previous execution,

and produces a new order of the operators, New_LS_Order by performing a descending order of the operators in the $Current\_LS\_Order$ or the $Initial\_LS\_Order$ based on the values of $LS\_Improvements\_Counter$. If no improvements occur, or if the same frequencies of improvements as the previous iteration are observed across all operators, then the initial order ($Initial\_LS\_Order$) is restored as the new sequence ($Current\_LS\_Order$) for the next iteration.

4.2.5. Shaking method

According to the diversification phase of the developed DA-GVNS algorithm, the adaptive intensified shaking method proposed in the work of Karakostas & Sifaleras (2022) has been adopted. This diversification method consists of three shaking local search operators (Relocate, Swap, and 2-Opt) and an adaptive re-ordering mechanism.

In the shaking phase, these operators are not fully applied sequentially in the incumbent route. Still, they are randomly performed for the number of times indicated by the intensity parameter k_{max}. The adaptive shaking re-ordering mechanism is similar to the one utilized in the improvement phase. The only difference is in the names of the parameters utilized:

- $l_{max} \rightarrow Sh_{max}$
- $Initial\_LS\_Order \rightarrow Initial\_Shaking\_Order$
- $Current\_LS\_Order \rightarrow Current\_Shaking\_Order$
- $LS\_Improvements\_Counter \rightarrow Shaking\_Improvements\_Counter$
- $New\_LS\_Order \rightarrow New\_Shaking\_Order$.

For a more thorough understanding and in-depth exploration of the adaptive re-ordering mechanisms employed in both the improvement and shaking phases, readers are directed to the work of Karakostas & Sifaleras (2022).

4.2.6. DA-GVNS

Herein, the overall solution method is presented to optimize the TSP route. The algorithm 2 summarizes the proposed DA-GVNS. This method receives an initial TSP solution S, the maximum shaking intensity level k_{max}, the number of all local search and shaking operators, l_{max} and Sh_{max} respectively, as well as the initial order of these operators.

In accordance with the operational functionality of the algorithm provided, the sequencing of the shaking operators is dynamically updated in each new iteration, as delineated by lines $4 - 10$. In the initial iteration, this updating process is initiated based on a predefined order. Subsequently, in ensuing iterations, the reordering is contingent upon the outcomes of the adaptive shaking mechanism. Furthermore, for each level of shaking intensity (line 12) and each selected shaking operator (lines $13 - 14$), a diversification phase is executed as outlined in line 15. Subsequently, the local search operators undergo reordering (lines $16 - 22$), and the solution derived from the shaking procedure is advanced to the improvement phase of the algorithm, as indicated in line 24. The newly obtained solution undergoes a comparative evaluation against the best solution identified hitherto (line 25). If it surpasses the current best, the latter is updated (line 26). The procedural sequence concludes upon reaching the upper execution time limit, and the overall best solution identified is reported (line 32).

Algorithm 2 Double Adaptive GVNS

```
procedure DA-GVNS($S, k_{max}, max\_time, l_{max}, Sh_{max}Initial\_LS\_Order, Initial\_Shaking\_Order$)
    $iteration = 1$
    while $time \leq max\_time$ do
        if $iteration = 1$ then
            $Current\_Shaking\_Order = Initial\_Shaking\_Order$
        else
            $New\_Shaking\_Order = Shaking\_Adaptive\_Mechanism(Current\_Shaking\_Order,$
                                                   $Initial\_Shaking\_Order, Sh_{max})$
            $Current\_Shaking\_Order = New\_Shaking\_Order$
        end if
        for $k \leftarrow 1$ to $k_{max}$ do
            for $i \leftarrow 1$ to $Sh_{max}$ do
                $l = Current\_Shaking\_Order(i)$
                $S^* = Shake(S, l)$
                if $iteration = 1$ then
                    $Current\_LS\_Order = Initial\_LS\_Order$
                else
                    $New\_LS\_Order = LS\_Adaptive\_Mechanism(Current\_LS\_Order,$
                                                   $Initial\_LS\_Order, l_{max})$
                    $Current\_LS\_Order = New\_LS\_Order$
                end if
                $S' = pVND(S^*, l_{max}, Current\_LS\_Order)$
                if $f(S') < f(S)$ then
                    $S \leftarrow S'$
                end if
            end for
        end for
        $iteration = iteration + 1$
    end while
    return $S$
end procedure
```

4.3. Second stage of the proposed solution method

4.3.1. Apply vehicle's load

The initial step of the second stage of the solution approach focuses on the configuration of the load of the vehicle while it moves from one node to another. The quantity delivered to each customer must be equal to his demand. Thus, an $|I| \times |I|$ matrix, $Load(:,:)$, is utilized to store the load of the vehicle while moving from a node $i \in I$ to a node $j \in I$. It is obvious that the load between the depot and the first customer node in the route will be equal to the total demand of the customers, while the load of the vehicle when it returns to the depot from the last customer node will be empty. The vehicle load will be decreased by the demand of each previous customer node in the scheduled route.

4.3.2. Assign speed levels

The next step in the second stage of the proposed solution method includes the assignment of speed levels to each active link of the scheduled route. To speed up this process, a speed level of $70km/h$ was allocated to all edges of the route. This value was selected by applying a ceiling function on the average of three different speed values $(45, 55, 105)$, which leads to $ceiling(68.33) = 69 \to 70km/h$. The first two values have been found to be the best choice in scheduling route planning considering fuel consumption (Karakostas et al., 2020, 2022), while the last was the best as obtained using a commercial solver in preliminary testing of the PTSPR model.

4.3.3. Fuel consumption calculation

This step focuses on the application of CMEM on the structured route to calculate the amount of fuel consumed by the vehicle at each edge. The expressions 27 of the mathematical model have been applied to achieve that. The calculated values are stored in a $N \times N$ matrix, $FuelConsumption(:,:)$.

4.3.4. Fuel levels calculation

The function of this step receives the scheduled route and the information produced in the previous step to calculate the fuel levels at each node location in the route. The calculated values are stored in a vector of size N, named $FuelLevel(:)$. The calculation of this step follows the formula: $FuelLevel(currentNode) = FuelLevel(previousNode) - FuelConsumption(previousNode, currentNode)$. For clarity, the fuel level at the depot is equal to the maximum level ($FuelLevel(depot) = UFL$).

4.3.5. Address refueling decisions

This step begins by scanning the fuel level at each node in the scheduled route. If a fuel level is less than or equal to LFL, refueling is required before reaching this node location. A logical variable is utilized to indicate in which location a refueling is required.

If refueling is required, a modified iterative variant of the Cheapest Insertion (CI) heuristic is applied. The developed insertion heuristic receives the location, $posI$, of the node i where the fuel level of the vehicle will be less than or equal to LFL with the current refueling schedule. It investigates which of the available fuel stations can be placed between the edge $(posI - 1, posI)$. Of these fuel stations, the most cost-effective is selected, considering the impact of this move on the total cost of the logistics system. If the placement of any fuel station on the selected edge is not valid, refueling is attempted on the previous edge, $(posI-2, posI-1)$. This process is executed in iterative mode until a valid refueling schedule can be addressed.

The $N \times N \times 2$ array, $Refueling(:,:,:)$, is used to store refueling schedules. $Refueling, (:,:,1)$ keeps the fuel station, while $Refueling(:,:,2)$ stores the amount of refueling. If no refueling is required, this array stores zero values. After selecting both the proper fuel station and its proper position on the route, a check for refueling requirements is performed. The process continues until there are no more refueling requirements.

4.3.6. Speed Optimization

After the completion of the iterative modified Cheapest Insertion method, an update of individual costs is performed. Then, the last component of the overall solution method is applied. This improvement component is the Speed Optimization algorithm (Demir et al., 2012; Karakostas et al., 2019a, 2022). From a procedural perspective, the Speed Optimization algorithm receives a solution, S, and a set with the available speed levels, L, and selects the best speed level for each edge of the route.

5. Computational results

This section is designed to furnish a comprehensive overview of the computational performance and behavior of the proposed solution method, coupled with a nuanced analysis of pertinent managerial insights for the focal problem. Initially, elucidation is provided on the computational environment and the generated problem benchmarks. Subsequently, a detailed computational analysis is undertaken to discern the contributions of each operator and to evaluate the impact of different diversification intensity levels and approaches. A comparative study ensues, juxtaposing the proposed DA-GVNS against alternative solution methods on symmetric TSPLib benchmarks, to assess the performance of the DA-GVNS. Following this, a dedicated computational study on the PTSPR generated benchmarks unfolds, wherein the proposed solution method is rigorously compared with the commercial solver, Gurobi. To conclude, the section culminates in the presentation of sensitivity analyses, offering valuable managerial insights.

5.1. Computing environment

The proposed solution methods were implemented using the Fortran programming language. Their execution was performed using the Intel Fortran compiler 18.0, under the optimization option -O3, on a laptop PC (Windows 10 Home 64-bit with an Intel Core i7-9750H CPU at 2.6 GHz and 16 GB RAM). The execution time of DA-GVNS was set at $40s$.

The proposed MILP model was implemented using the Gurobi-Python API v.10.0.0. In the case of the Gurobi solver, an initial time limit of 2 hours was defined for execution. However, if the commercial solver failed to produce any viable integer solution within the initial time limit, the execution time was extended to 5 hours. Furthermore, the Gurobi solver was configured with the following parameter settings: the “Aggregate” parameter was set to a value of 2, the “Cuts” parameter was set to 3, the “NodefileStart” parameter was set to 0.5, the “Threads” parameter was set to 1, the “NodeMethod” parameter was set to 2, and the “MIPFocus” parameter was set to 1.

5.2. Problem instances

To facilitate a comprehensive computational analysis, we generated 20 new benchmark instances using a randomization process. Each instance is denoted as "$TSP - |I| - |J|$", where "$|I|$" represents the total number of operational nodes (comprising the depot and customers), and "$|J|$" indicates the count of available fuel stations. Crucial details about each instance, including these parameters, are explicitly stated in the initial line of the respective instance file. The structure of each instance file adheres to the following format:

- The first line provides information about the total number of operational nodes and available fuel stations.
- Subsequently, the second line furnishes the coordinates of the depot.
- The subsequent $|I| - 1$ lines contain the coordinates of individual customers.
- The next $|J|$ lines offer the coordinates of available fuel stations along with their respective fuel prices.
- The demand for each operational node (excluding the depot) is presented in the following $|I| - 1$ lines.
- The last line of each instance file specifies the vehicle capacity.

For the generation of instance data, the coordinates of nodes were systematically generated from a uniform distribution within the interval $[0, 100]$. Fuel prices were also randomly drawn from a uniform distribution in the range $[1.86, 2]$. The demand of each customer was generated using a normal distribution with a mean value ("mean") selected from a uniform distribution over the interval $[5, 15]$ and a standard deviation ("sd") generated from a uniform distribution over the interval $[0, 5]$. Lastly, the vehicle capacity was set equal to the total demand of all customers in each instance. The new benchmark set is publicly available at `https://sites.uom.gr/sifalera/benchmarks.html`.

5.3. Neighborhoods contribution

This section focuses on the analysis of the contribution of each local search operator and their combinations to the performance of the DA-GVNS for the solution of TSP instances. To achieve that, each operator or combination examined was deployed in an adaptive pVND within a best-improvement search strategy and a fixed adaptive intensified shaking method ($k_{max} = 8$) from the literature (Karakostas & Sifaleras, 2022). For each operator or combination of operators, a distinct version of the DA-GVNS was instantiated. These formed versions of DA-GVNS were employed to address all generated problem instances. The reported percentage performance deviations represent the averaged values obtained from 20 independent runs for each version of DA-GVNS. Figure 9 illustrates the contribution of the six best-performing combinations. The reported gaps denote the average percentage performance deviation between the solutions produced by the construction heuristic (NNH) and those derived from each operator or combination, considering all generated problem instances.

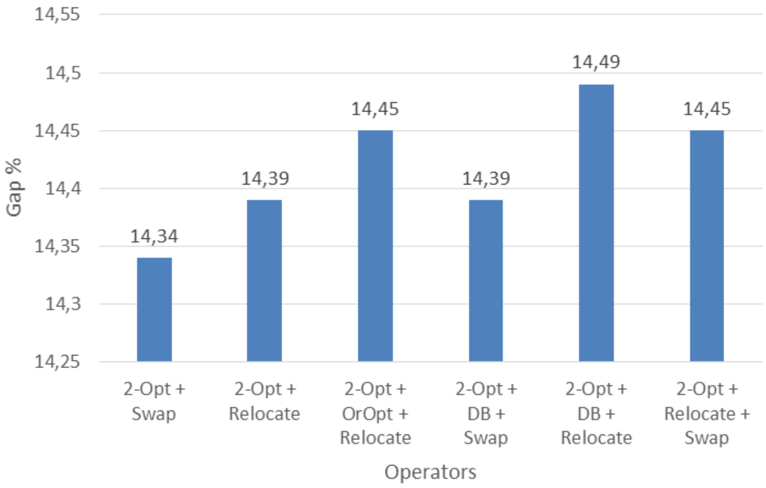


Figure 9: Contribution of local search operators

The combination of $2-Opt$, $Double-Bridge(DB)$, and $Relocate$ led to the best performance, and they were selected to be embedded in the adaptive pVND of the DA-GVNS. However, it was noticed that by applying a limit on the depth of search of the DB operator, further improvements could be achieved. Eight limits were tested and their results are presented in Table 5. The second column displays the results of the NNH, while the subsequent eight columns showcase the outcomes obtained by the proposed DA-GVNS under the different depth limits in the DB operator. Additionally, the reported percentage deviations denote the average cost difference between the solutions reported by the DA-GVNS and those achieved by the NNH.

Table 5: TSP objective values due to different iteration limits of DB operator

Limits		**30**	**40**	**50**	**60**	**70**	**80**	**90**	**100**
Instances	NNH								
TSP_15-5	382361.95	342524.44	342524.44	342524.44	342524.44	342524.44	342524.44	342524.44	342524.44
TSP_25-5	487364.45	413455.03	413455.03	413455.03	413455.03	413455.03	413455.03	413455.03	413455.03
TSP_30-5	627282.29	461377.84	461377.84	461377.84	461377.84	461377.84	461377.84	461377.84	461377.84
TSP_50-5	672069.38	557810.81	557810.81	557810.81	557810.81	557810.81	557810.81	557810.81	557810.81
TSP_60-7	770221.86	630181.06	629083.06	629083.06	630181.06	629083.06	629083.06	629083.06	629083.06
TSP_85-8	865519.43	724531.19	726725.63	730276.19	726725.63	730276.19	730276.19	728904.38	726421.19
TSP_100-5	929140.65	786118.25	786118.25	791068.00	787256.38	791068.00	791068.00	791068.00	786118.31
TSP_100-10	877334.55	766911.63	766911.63	766205.5	769814.79	766911.63	766911.63	766877.88	766877.88
TSP_120-10	930409.07	754949.88	752402.69	752402.69	754949.88	752402.69	752402.69	754949.88	754949.88
TSP_150-10	1212181.03	956616.88	959567.19	960980.19	958254.25	976451.94	961214.13	976370.35	962758.42
TSP_180-10	1226097.03	1061925.59	1048815.25	1058317.99	1062862.75	1057517.13	1068073.50	1055985	1056697.25
TSP_200-10	1326034.08	1063774.50	1042708.63	1065813.50	1045665.12	1056536.63	1054132.88	1059294.88	1063373.13
TSP_250-15	1563876.1	1251372.87	1262667.38	1255730.13	1267016.75	1256106.63	1260179.25	1253221	1253294.63
TSP_300-10	1548535.54	1326912.87	1334879.25	1323111.36	1317131.25	1330716.63	1326577.09	1320835.38	1337534.00
TSP_300-15	1657858.99	1335881.00	1343165.34	1358638	1337177.23	1356147.00	1340949.13	1362002.88	1358126.86
TSP_500-15	2071805.88	1728892.72	1724693.75	1710513.38	1727854.97	1731821.88	1711083.95	1741931.88	1739843.88
TSP_550-20	2094172.87	1829014.00	1829445.13	1829605.35	1815240.75	1817877.60	1809373.32	1823656.86	1826993.21
TSP_800-25	2551755.50	2285839.50	2311880.66	2287869.75	2295034.75	2302308.92	2286896.67	2296170.23	2289152.88
TSP_950-25	2859461.53	2609325.44	2621502.25	2622188.75	2625357.95	2691807.13	2623086.60	2670185.68	2640867.87
$TSP\_1000-25$	2930022.80	2700425.74	2690389.20	2677735.61	2685292.44	2675633.75	2674790.94	2684273.95	2701768.95
Average	1379175.25	1179392.06	1180306.17	1179735.38	1179049.20	1184891.75	1178063.36	1184498.97	1183451.48
	Gap %	14.49	14.42	14.46	14.51	14.09	**14.58**	14.12	14.19

The computational results of testing different limits on DB revealed that a limit of 80 iterations per call of the DB operator can lead to better solutions compared to other values. After the configuration of the improvement phase of the developed DA-GVNS, we focused on the computational analysis of the diversification phase. Due to the fact that the improvement phase in this work is differentiated from the one considered in the work of Karakostas & Sifaleras (2022), shaking intensity levels, intensity strategies, and search strategies were also investigated.

To clarify, three search strategies, the best improvement, the first improvement, and the adaptive search strategy, were investigated. Based on the intensity of shaking, two intensity strategies were evaluated, one that considers fixed k_{max} values and one in which the k_{max} value is received as proportional to the number of the nodes. Thus, the following k_{max} values were considered:

- fixed k_{max} values: $(2, 4, 8, 10, 12, 15)$,
- k_{max} values proportional to $|N|$: $(0.05 \times |N|, 0.1 \times |N|, 0.15 \times |N|, 0.2 \times |N|, 0.5 \times |N|)$.

Table 6 summarizes the average best objective values of all instances, achieved by employing the most efficient combinations of search and shaking strategies.

Table 6: Best found objective values using different shaking and search strategies

	Best Improvement		First Improvement		Adaptive Search	
	$k_{max} = 2$	$k_{max} = 0.05 \cdot \lvert N \rvert$	$k_{max} = 10$	$k_{max} = 0.05 \cdot \lvert N \rvert$	$k_{max} = 15$	$k_{max} = 0.01 \cdot \lvert N \rvert$
Objective value	1179406.01	1184391.08	1160607.07	1207808.6	**1131372.36**	1171545.8
Gap %	14.48	14.12	15.85	12.43	**17.97**	15.05

The consideration of the k_{max} values received as proportional to the number of nodes led to worse solutions than in the case of fixed shaking intensity levels. Although the best improvement search strategy performed better in small instances, the first improvement search strategy presented the best results in the case of large instances. However, the adaptive search strategy performed significantly better than other approaches, leading to a solution almost 18% better than the initial solution of the construction heuristic.

5.4. Computational analysis on PTSPR instances

This section presents the computational results of PTSPR instances using the proposed solution method and the Gurobi solver. These results are provided in Table 7. In the primary computational investigation, the study incorporates an 85-liter fuel tank capacity, along with a lower refueling threshold set at one-fourth of the upper fuel limit (UFL). With respect to the remaining parameters, in summary, the problem-specific parameters delineated in Tables 2 and 3, along with the technical parameters, namely $k_{max} = 15$ and a limitation of 80 iterations per invocation of the DB, have been taken into account. Furthermore, considering the adaptive search strategy, the best improvement is employed when the number of cities is less than or equal to 600, while the first improvement strategy is applied otherwise. The second column contains the objective values obtained using the DA-GVNS combined with the modified CI heuristic, while the third one summarizes the results produced using the overall method (DA-GVNS - CI - Speed Optimization). The value of each cell in these two columns is the average of the objective values obtained by each method in 50 independent runs. The consideration of the overall proposed solution method produces 28.51% better solutions than DA-GVNS combined with CI. The last column provides the objective values obtained by using the Gurobi solver. The abbreviation "NIS" indicates that no integer solution was obtained using the Gurobi solver, whereas the abbreviation "OOM" signifies an out-of-memory error encountered during the execution of the commercial solver.

Owing to the substantial computational complexity associated with the new optimization problem, the commercial solver was capable of generating feasible integer solutions for only seven out of the total 20 instances of the problem. Among these seven instances, four were successfully solved within the initial execution time limit, while an extension of the execution time limit was necessary for the remaining three instances. However, despite allowing a maximum execution time of 5 hours, the commercial solver failed to produce any feasible integer solutions for the "TSP-25-5" and "TSP-100-5" problem instances. Additionally, an out-of-memory error occurred for the remaining problem instances. Considering these limitations, it becomes crucial to explore alternative approaches, such as the heuristic method employed in this study. Although the commercial solver surpasses the heuristic in terms of solution quality, it is noteworthy that the heuristic exhibits commendable efficiency, requiring significantly less execution time to generate solutions. This indicates the potential of the heuristic to efficiently handle large-scale instances characterized by substantial computational complexity.

Furthermore, it is important to note that while the commercial solver consistently outperforms the heuristic approach in terms of solution quality, the observed gap in solution quality is significantly mitigated when considering the best solutions found by the heuristic. Although the commercial solver may provide superior solutions for the limited instances it can solve, the heuristic demonstrates its potential by consistently delivering competitive and satisfactory solutions within a fraction of the execution time. By carefully analyzing and leveraging the best solutions obtained from the heuristic, valuable insights can be gained, enabling further improvements in solution quality. Therefore, despite the limitations of the heuristic, its ability to yield viable and competitive solutions cannot be disregarded, making it a valuable tool in addressing the computational complexities of the optimization problem at hand.

Table 7: PTSPR objective values

Instance	$DA-GVNS-CI_{avg}$	$2S-DA-GVNS_{avg}$	$2S-DA-GVNS_{best}$	$2S-DA-GVNS_{worst}$	$2S-DA-GVNS_{sd}$	$Gurobi$
TSP_15-5	285.66	210.41	203.75	270.31	20.17	182.32
TSP_25-5	598.82	428.73	418.73	563.36	34.75	NIS
TSP_30-5	509.30	351.26	340.36	476.57	38.33	303.90
TSP_50-5	708.52	496.74	474.93	729.07	67.26	366.22
TSP_60-7	1152.86	504.37	498.55	578.88	19.96	522.30
TSP_85-8	1051.72	643.60	624.89	807.18	51.72	550.73
TSP_100-5	1014.36	687.15	668.70	937.97	63.56	NIS
TSP_100-10	1301.16	657.17	645.52	762.02	35.30	610.18
TSP_120-10	993.63	682.25	670.45	818.75	40.43	12382.15
TSP_150-10	1326.87	880.35	837.73	1295.81	118.78	OOM
TSP_180-10	1362.75	990.92	947.83	1252.59	99.97	OOM
TSP_200-10	1367.78	965.97	931.92	1198.90	72.71	OOM
TSP_250-15	1655.61	1192.62	1145.15	1434.99	87.64	OOM
TSP_300-10	1821.74	1305.13	1258.69	1604.41	116.31	OOM
TSP_300-15	1761.00	1318.56	1290.17	1518.81	62.14	OOM
TSP_500-15	2469.73	1874.83	1863.31	2011.15	39.48	OOM
TSP_550-20	2503.22	1891.19	1857.81	2115.47	78.06	OOM
TSP_800-25	3129.10	2404.96	2365.53	2687.53	98.92	OOM
TSP_950-25	3452.54	2690.34	2677.01	2899.23	53.31	OOM
$TSP\_1000-25$	3838.11	2917.95	2875.27	3326.06	129.45	OOM
Average	1615.22	1154.72	1129.81	1364.45	66.36	-

According to the execution CPU time of the developed heuristic approach, the first method required 40.05s for the solution of the largest instance, while the second required

61.19s respectively (fixed 40s for the DA-GVNS, 0.05s for the CI and 21.14s for the Speed Optimization). In our investigation of the analytical solutions for the five problem instances, focusing specifically on the comparative performance between the commercial solver and the heuristic method, we have observed a notable improvement achieved through the utilization of lower speed levels. Specifically, our analysis reveals that the average speed attained by the commercial solver across all instances is recorded as 46.56 km/h, while the corresponding average speed obtained by the heuristic approach stands at 50.69 km/h. This discrepancy in speed levels translates to a significant disparity in fuel consumption requirements. Based on the solutions provided by the Gurobi solver, an average of 428.61 liters of fuel needs to be refueled, whereas the heuristic approach necessitates an average refueling quantity of 526.93 liters. This increase of 8.87% in the average speed directly contributes to a subsequent rise of 22.94% in average fuel consumption. Consequently, this increase in fuel consumption not only results in higher costs associated with emissions taxation but also leads to an elevated expenditure on refueling.

Herein, we investigate the potential improvements in the quality of the solution that can be achieved by extending the execution time limit of the $2S-DA-GVNS$. Specifically, we explore four distinct execution time limits (60s, 120s, 300s, 360s) for the $DA-GVNS$ method, focusing on the five small and medium-sized problem instances. Table 8 presents the average, best, and worst objective values, together with the corresponding standard deviation, derived from 50 independent runs of the $2S-DA-GVNS$ under these varying time limits. By extending the time limit to 381.19s (360s for DA-GVNS and 21.19s for the CI within Speed Optimization), a significant improvement in the efficiency of the $2S-DA-GVNS$ compared to the Gurobi solver appeared; despite the latter requires considerably longer execution time (2-5 hours).

Table 8: Impact of execution time adjustments

Instance	Gurobi	Execution CPU Time limits				
			60 s	120s	300s	360s
TSP_15-5	182.32	Average	209.66	205.82	203.40	188.64
		Best	203.75	203.75	182.32	**182.32**
		Worst	243.87	243.87	219.65	219.65
		SD	10.94	7.01	9.60	10.47
TSP_30-5	303.9	Average	333.76	327.40	322.60	319.07
		Best	324.67	303.90	303.90	**303.90**
		Worst	340.36	340.36	340.34	340.34
		SD	7.82	9.89	9.28	10.49
TSP_50-5	366.22	Average	479.61	479.45	478.10	477.79
		Best	474.93	474.93	474.93	474.93
		Worst	490.06	490.06	490.06	482.46
		SD	5.69	4.31	4.60	3.69
TSP_85-8	550.73	Average	627.31	626.32	562.51	548.79
		Best	624.89	624.89	559.03	**542.08**
		Worst	649.84	637.61	566.37	563.50
		SD	6.33	3.41	3.05	8.72
TSP_100-10	610.18	Average	609.33	603.70	579.26	575.08
		Best	580.05	570.34	562.75	**562.75**
		Worst	694.65	694.65	580.05	580.05
		SD	33.82	34.56	3.41	7.77

Table 9 provides the total and individual costs of the best-found solution for each PTSPR. The asterisk symbol serves as an indicator that the best solution for a particular instance was obtained through the utilization of the commercial solver.

Table 9: Total and individual costs of the best-found solution

Instance	**Total Cost**	**Driver Wage Cost**	**Emissions Tax Cost**	**Refueling Cost**
$TSP\_15 - 5^*$	182.32	68.42	67.77	46.15
$TSP\_25 - 5$	418.63	103.06	131.54	184.03
$TSP\_30 - 5^*$	303.90	101.68	93.30	108.92
$TSP\_50 - 5^*$	366.22	114.46	107.62	144.14
$TSP\_60 - 7$	498.55	134.56	150.35	213.65
$TSP\_85 - 8$	542.08	177.79	128.69	235.6
$TSP\_100 - 5$	668.70	176.05	191.49	301.16
$TSP\_100 - 10$	562.75	194.94	141.35	226.46
$TSP\_120 - 10$	670.45	176.14	189.07	189.07
$TSP\_150 - 10$	837.73	217.58	220.84	399.31
$TSP\_180 - 10$	947.83	242.63	239.62	465.57
$TSP\_200 - 10$	931.92	238.91	223.65	469.35
$TSP\_250 - 15$	1145.15	287.20	295.17	562.78
$TSP\_300 - 10$	1258.69	312.51	323.31	622.87
$TSP\_300 - 15$	1290.17	319.01	316.97	654.19
$TSP\_500 - 15$	1863.31	430.04	458.85	974.42
$TSP\_550 - 20$	1857.81	428.02	431.74	998.04
$TSP\_800 - 25$	2365.53	522.40	586.99	1256.14
$TSP\_950 - 25$	2677.01	527.87	667.09	1437.05
$TSP\_1000 - 25$	2875.27	613.84	717.60	1543.83

5.5. Sensitivity analysis

The dimensions of the fuel tank and the refueling policy are recognized as two pivotal parameters in the proposed model. Consequently, additional computational analysis is deemed necessary to assess the influence of potential variations in these parameters on the structure and cost of the logistics system. This examination aims to provide a deeper understanding of the implications and repercussions associated with fluctuations in the values of these critical parameters. It is crucial to emphasize that the conducted sensitivity analyses were performed exclusively using the heuristic approach, specifically relying on the best solutions found and reported by the algorithm, under a maximum execution time limit of $61.19s$. This choice was motivated by the heuristic's remarkable capability to generate solutions for problem instances of varying sizes, including those characterized as large-scale, which hold practical significance in the context of the study.

According to the fuel tank, we considered three additional sizes, $(105L, 125L, 150L)$, while for the refueling, the following two refueling policies were examined, $LFL = \frac{1}{3} \times UFL$, and $LFL = \frac{1}{2} \times UFL$.

Tables 10, 11, 12, and 13 provide a comprehensive overview of the alterations observed in the total costs, fuel costs, emission taxation costs, and driver wages costs, associated with different combinations of fuel tank sizes and refueling policies.

Table 10: Average values of the best-found total costs (€) for all PTSPR instances

Fuel Tank Size (L)	$\boldsymbol{LFL = \frac{1}{4} \times UFL}$	$\boldsymbol{LFL = \frac{1}{3} \times UFL}$	$\boldsymbol{LFL = \frac{1}{2} \times UFL}$
85	1129.81	Infeasible	Infeasible
105	1238.05	1180.68	1213.73
125	1178.22	1178.16	1181.74
150	1138.83	1137.04	1145.18

Table 11: Average values of the best-found refueling costs (€) for all PTSPR instances

Fuel Tank Size (L)	$\boldsymbol{LFL = \frac{1}{4} \times UFL}$	$\boldsymbol{LFL = \frac{1}{3} \times UFL}$	$\boldsymbol{LFL = \frac{1}{2} \times UFL}$
85	569.92	Infeasible	Infeasible
105	638.24	604.12	621.47
125	601.66	602.62	606.77
150	573.81	573.71	578.53

Table 12: Average values of the best-found emissions taxation costs (€) for all PTSPR instances

Fuel Tank Size (L)	$\boldsymbol{LFL = \frac{1}{4} \times UFL}$	$\boldsymbol{LFL = \frac{1}{3} \times UFL}$	$\boldsymbol{LFL = \frac{1}{2} \times UFL}$
85	290.32	Infeasible	Infeasible
105	333.23	316.22	330.29
125	316.17	316.34	313.23
150	311.08	308.14	309.39

Table 13: Average values of the best-found driver wages costs (€) for all PTSPR instances

Fuel Tank Size (L)	$LFL = \frac{1}{4} \times UFL$	$LFL = \frac{1}{3} \times UFL$	$LFL = \frac{1}{2} \times UFL$
85	269.56	Infeasible	Infeasible
105	266.08	260.2	261.97
125	260.39	259.19	261.74
150	253.94	255.2	257.21

Initially, it is noteworthy that the cost of refueling significantly contributes to the overall system cost. With a specific focus on the impact of alternative fuel tank sizes on the total cost, the following observations emerge. Under the influence of the main refueling policy ($LFL = \frac{1}{4} \cdot UFL$), employing a fuel tank capacity of 105 liters instead of 85 liters results in a notable increase of approximately 9.6% in the total cost. However, employing even larger fuel tanks appears to alleviate this cost escalation. Specifically, an increase of 4.28% is observed when employing a fuel tank capacity of 125 liters, while a marginal increase of 0.8% is observed when transitioning from an 85-liter fuel tank to a 150-liter fuel tank. These observations can be attributed directly to fluctuations in fuel consumption and, consequently, the required refueling quantities. Moreover, employing larger fuel tanks prompts the solution methodology to adopt higher speed levels, thereby achieving a subtle balance between costs related to fuel consumption and driver wages.

When considering the second refueling policy ($LFL = \frac{1}{3} \cdot UFL$), a notable observation emerges: employing an 85-liter fuel tank renders the design of feasible logistic plans unattainable. This observation underscores the pivotal role of adopting an analytical fuel consumption model that accounts for various vehicle characteristics and integrates it into refueling decisions. Specifically, in this particular case, the absence of a refueling decision would result in the determination of a routing plan with a total fuel consumption amount, negating the realistic necessity of approaching fuel stations to meet the evident refueling requirement. Similarly, associating fixed fuel consumption rates with each distance (link) would fail to capture the realistic limitations arising from vehicle-specific attributes.

Continuing with the observations made under the influence of the second refueling policy, employing a 125-liter fuel tank instead of a 105-liter fuel tank leads to a marginal reduction in total cost. However, utilizing a 150-liter fuel tank presents substantial cost benefits compared to the alternatives of 105-liter and 125-liter fuel tanks (yielding cost reductions of 3.7% and 3.5% respectively). These findings underscore the potential cost-saving advantages of employing larger fuel tanks, as they enable more efficient utilization of available resources, thereby optimizing the overall logistics planning process. More specifically, the utilization of larger fuel tanks results in reduced refueling requirements, as lower quantities of fuel need to be replenished along the route. This reduction in refueling needs not only saves on fuel costs but also enables the solution methodology to apply higher speed levels in certain links of the route. By operating at higher speeds, the driver wage costs can be mitigated, as the time required for transportation is reduced. Consequently, the combination of reduced fuel costs and lower driver wage costs contributes to overall cost savings when larger fuel tanks are employed. These findings underscore the importance of considering the interplay between fuel tank sizes, refueling strategies, and their impact on both fuel-related costs and driver wage expenses for efficient logistics planning.

Similarly, analogous observations arise when examining the impact of the third, more stringent, refueling policy ($LFL = \frac{1}{2} \cdot UFL$). It is observed that employing an 85-liter fuel tank renders the generation of feasible logistic plans unachievable, underscoring the necessity for alternative approaches. Notably, the adoption of larger fuel tanks is associated with substantial cost benefits. These findings further reinforce the notion that considering larger fuel tank sizes plays a crucial role in optimizing the logistics planning process and achieving cost efficiency.

Turning our attention to the influence of fluctuations in refueling policies on overall costs for each fuel tank scenario, it is apparent that the utilization of the second refueling policy aligns more favorably with the use of a 105-liter or 125-liter fuel tank. Conversely, adopting more relaxed refueling policies proves more advantageous when employing a 150-liter fuel tank.

To gain deeper insights into these observations, Tables 14 and 15 offer a comprehensive analysis of the number of refueling stops and the corresponding total refueling quantities for each problem case, accounting for different fuel tank sizes and refueling policies. The symbol "X" in these Tables signifies the absence of a feasible solution for the given combination of fuel tank capacity and refueling policy.

Table 14: Number of intermediate refueling stops for each problem case

	$\boldsymbol{LFL = \frac{1}{4} \cdot UFL}$				$\boldsymbol{LFL = \frac{1}{3} \cdot UFL}$				$\boldsymbol{LFL = \frac{1}{2} \cdot UFL}$			
Instance	$85L$	$105L$	$125L$	$150L$	$85L$	$105L$	$125L$	$150L$	$85L$	$105L$	$125L$	$150L$
TSP_15-5	1	1	1	1	X	1	1	1	X	1	1	1
TSP_25-5	3	3	2	1	X	2	2	1	X	3	2	1
TSP_30-5	2	2	2	1	X	2	2	1	X	2	2	1
TSP_50-5	3	3	3	2	X	3	3	2	X	3	3	2
TSP_60-7	3	3	3	2	X	3	3	2	X	3	3	2
TSP_85-8	4	4	3	3	X	4	3	3	X	4	3	3
TSP_100-5	4	5	3	3	X	3	3	3	X	5	3	3
TSP_100-10	4	4	4	3	X	3	3	3	X	4	4	3
TSP_120-10	4	5	3	3	X	4	4	3	X	5	4	3
TSP_150-10	5	6	5	4	X	5	5	3	X	6	5	4
TSP_180-10	6	7	5	4	X	5	5	4	X	7	5	4
TSP_200-10	6	7	5	4	X	5	5	4	X	6	5	4
TSP_250-15	7	8	6	5	X	6	6	5	X	8	6	5
TSP_300-10	8	9	7	6	X	7	7	6	X	8	7	6
TSP_300-15	8	8	7	5	X	6	7	5	X	8	7	6
TSP_500-15	12	12	9	7	X	9	9	8	X	12	9	7
TSP_550-20	12	12	10	8	X	10	10	8	X	12	10	8
TSP_800-25	15	15	12	10	X	12	12	10	X	15	12	10
TSP_950-25	17	17	14	11	X	14	14	11	X	16	14	11
$TSP\_1000-25$	18	18	15	12	X	15	15	12	X	18	15	12

Table 15: Total refueling quantities for each problem case (in liters)

	$\boldsymbol{LFL = \frac{1}{4} \cdot UFL}$				$\boldsymbol{LFL = \frac{1}{3} \cdot UFL}$				$\boldsymbol{LFL = \frac{1}{2} \cdot UFL}$			
Instance	85L	105L	125L	150L	85L	105L	125L	150L	85L	105L	125L	150L
TSP_15-5	32.61	36.35	44.31	44.31	X	44.31	44.31	44.31	X	36.35	44.31	44.31
TSP_25-5	94.07	124.83	85.73	44.26	X	84.07	78.76	44.26	X	124.83	84.07	44.26
TSP_30-5	71.49	69.06	86.53	49.48	X	90.74	92.4	50.52	X	69.06	93.41	50.52
TSP_50-5	108.06	124.41	149.73	112.34	X	149.73	149.73	112.42	X	116.38	149.73	109.23
TSP_60-7	109.6	124.16	138.6	114.7	X	142.2	138.6	112.57	X	124.16	138.6	109.06
TSP_85-8	161.98	162.84	151.61	176.13	X	197.02	151.61	171.23	X	163.92	149.01	174.01
TSP_100-5	156.52	207.84	138.04	175.46	X	139.31	139.31	173.46	X	201.66	139.31	169.58
TSP_100-10	152.79	168.34	195.45	168.21	X	142.85	142.7	170.48	X	168.34	201.48	170.75
TSP_120-10	160.44	210.16	145.12	176.12	X	191.14	199.91	181.15	X	206.22	196.45	172.08
TSP_150-10	202.55	251.92	252.67	239.47	X	245.97	249.55	180.4	X	250.68	245.07	239.47
TSP_180-10	240.63	299.86	250.75	229.67	X	253.95	251.31	240.41	X	295.23	248.5	238.05
TSP_200-10	247.82	297.75	253.64	246.47	X	255.11	260.43	235.8	X	257.18	253.03	233.7
TSP_250-15	289.07	336.35	304.3	300.63	X	305.58	303.21	305.68	X	344.38	308.25	302.6
TSP_300-10	328.5	397.01	354.45	367.84	X	356.07	360.4	362.15	X	342.66	353.89	363.83
TSP_300-15	336.47	348.63	353.96	307.49	X	313.91	351.81	308.84	X	344.65	362.92	367.09
TSP_500-15	509.47	543.94	479.52	453.78	X	488.73	476.96	510.31	X	532.05	484.12	457.23
TSP_550-20	516.51	542.52	536.44	516.41	X	540.71	536.33	516.25	X	542.01	531.47	583.67
TSP_800-25	656.39	704.18	677.24	662.21	X	672.33	665.76	663.49	X	700.31	671.89	599.31
TSP_950-25	753.38	814.95	792.49	753.72	X	795.43	794.94	750.08	X	764.48	791.98	752.78
$TSP\_1000-25$	806.37	874.42	857.29	820.4	X	866.93	871.2	819.58	X	868.75	854.16	822.58
Average	296.74	331.98	312.39	297.96	X	313.8	312.96	297.67	X	322.67	315.08	300.21

It is important to emphasize that the aforementioned observations are contingent upon the underlying assumptions and specific characteristics of the problem at hand. Nevertheless, these findings demonstrate the efficacy of the proposed method in aiding logistics decision-makers in designing efficient logistic plans by thoroughly examining various alternative scenarios. By investigating the impact of different fuel tank sizes and refueling policies, the proposed method offers valuable insights that facilitate informed decision-making and enhance the overall effectiveness and cost-efficiency of logistics operations. These observations underscore the significance of considering alternative scenarios and refining logistics strategies to optimize resource utilization and achieve sustainable transportation solutions in real-world logistics environments.

6. Comparative study on symmetric TSPLib instances

This section presents a comparative analysis between the fundamental component of our proposed solution methodology, the DA-GVNS, and other efficient heuristic algorithms documented in the literature for solving symmetric instances derived from the TSPLib dataset. The results produced by our proposed solution method are compared with those achieved by several heuristic algorithms, including a DA-GVNS (Karakostas & Sifaleras, 2022), an improved VNS (iVNS) (Hore et al., 2018), another VNS (Bingüler & Bulkan, 2015), as well as two nature-inspired heuristics, a Discrete Symbiotic Organisms Search algorithm (DSOS) (Ezugwu & Adewumi, 2017) and a combinatorial Bee algorithm (cBA) (Sahin, 2023). The average objective values obtained by each solution method under consideration are presented in Table 16. It is noteworthy that the reported results for VNS Bingüler & Bulkan (2015) correspond to the best-found values.

Table 16: Comparison between the proposed method and other solution approaches

Instance	Optimal	Proposed method	DA-GVNS	iVNS	VNS	DSOS	cBA
a280	2579	2674	2644				
ali535	202339	214198	217371				
att48	10628	10628	10628		10628		
att532	27686	29268	28924				
bayg29	1610	1610	1610		1610		
bays29	2020	2020	2020	2020	2020		
berlin52	7542	7542	7542	7544.36	7542	7542.60	7544
bier127	118282	118660	120077	119006.39			
brazil58	25395	25395	25395	25592.72			
brg180	1950	1972	1974				
burma14	3323	3323	3323				
ch130	6110	6148	6166	6153.72	6368		
ch150	6528	6570	6608	6644.95			6554
d198	15780	15913	15943	16079.28			15889
d493	35002	36515	36769				35864
d657	48912	52525	52209				
d1291	50801	55199	55269	56095.33			
d1655	62128	66860	67551	70337.23			
d2103	80450	83643	84088				
dantzig42	699	699	699	699			
dsj1000	18659688	19925278	20212596				
eil51	426	426	427	428.98	426	427.90	
eil76	538	538	539	552.57	538	547.40	
eil101	629	632	637	648.27	644	650.60	

Continued on next page

Table 16 – continued from previous page

Instance	Optimal	Proposed method	DA-GVNS	iVNS	VNS	DSOS	cBA
fl417	11861	12093	12129	12183.14			
fl1400	20127	20983	22408	21085.98			
fl1577	22249	23990	24350				
fl3795	28772	31011	35948				
fnl4461	182566	201459	216965				
fri26	937	937	937	937	937		
gil262	2378	2472	2479	2501.86			2427
gr17	2085	2085	2085	2085			
gr21	2707	2707	2707	2707			
gr24	1272	1272	1272	1272	1272		
gr48	5046	5046	5046	5046	5046		
gr96	55209	55413	55448		55774		
gr120	6942	7007	6998		7127		
gr137	69853	70285	70233				
gr202	40160	41163	41411		43668		
gr229	134602	137861	137400				
gr431	171414	179891	181904				
gr666	294358	311923	314951				
hk48	11461	11461	11461		11461		
kroA100	21282	21286	21286	21695.79	21624	21409.50	21285
kroB100	22141	22167	22172	22140.20	22715	22339.20	22212
kroC100	20749	20786	20775	20809.29	20818	20881.60	20855
kroD100	21294	21370	21297	21490.62	21621	21439.10	21415
kroE100	22068	22124	22149	22193.80	22424	22231.10	22132
kroA150	26524	26758	26829	26947.17			26734
kroB150	26130	26462	26499	26537.04	28700		26302
kroA200	29368	29920	30002	30339.67			29533
kroB200	29437	30124	30203	30453.22	64871		29837
lin105	14379	14409	14379	14395.64	14596		14396
lin318	42029	43612	43664	43964.93		42972.42	42965
nrw1379	56638	60140	60484				
p654	34643	35009	35569				
pa561	2763	2904	2927				
pcb442	50778	52959	53020	50800.24			51836
pcb1173	56892	61214	61979	63435.95			
pcb3038	137694	149146	156236	154565.40			
pr76	108159	108159	108183	108159	108644		
pr107	44303	44693	44303	44314.92	46071	44445.10	
pr124	59030	59080	59090	59051.82	59813	59429.10	59210
pr136	96772	97336	97827	97985.84	101477	97673.20	
pr144	58537	58537	58537	58563.97	59834	58817.10	58688
pr152	73682	73921	74070	73855.11	78294	74785.70	
pr226	80369	80879	80749	80514.64	88494		81100
pr264	49135	50127	50143	51197.14		52798.90	
pr299	48191	50300	50113	50373.12		50335.20	48694
pr439	107217	111880	113495	111771.20			
pr1002	259045	278881	277329	280563.90		278381.51	277525
pr2392	378032	404981	414866			425431.78	
rat99	1211	1215	1218	1241.26		1228.37	
rat195	2323	2376	2387	2453.81			2352
rat575	6773	7255	7284	7362.51		7117.32	
rat783	8806	9343	9458	9707.36		9102.67	
rd100	7910	7911	7952	7918.36	8022		
rd400	15281	15971	15899	16250.21			
rl1304	252948	278501	282173				
rl1323	270199	287821	295999	295611.20			

Continued on next page

Table 16 – continued from previous page

Instance	Optimal	Proposed method	DA-GVNS	iVNS	VNS	DSOS	cBA
rl1889	316536	348928	344692				
rl5915	565530	630953	688397				
rl5934	556045	616836	660342				
si175	21407	21425	21434				
si535	48450	48848	49505				
si1032	92650	93054	92909				
st70	675	675	675	677.11		679.20	677
swiss42	1273	1273	1273	1273			
ts225	126643	127256	126990				
tsp225	3916	4037	4024				
u159	42080	42595	50778	42467.61			
u574	36905	39261	46286	39629.11			
u724	41910	44440	46112	45729.71			
u1060	224094	241639	241646				
u1432	152970	163439	166283				
u1817	57201	62391	62140				
u2152	64253	70265	70628				
u2319	234256	242826	243401	262595.60			
ulysses16	6859	6859	6859		6859		
ulysses22	7013	7013	7013		7013		
vm1084	239297	254445	257842				
vm1748	336556	357650	371452	366757.80			

Comparing the outcomes obtained by our $DA - GVNS$ with those achieved by other solution methods, reported in the literature, our method yielded improvements of 1.29% (1.56%), 1.06% (1.23%), 5.54% (5.7%), and 0.7% (0.9%) over the solutions presented by the $DA - GVNS^*$, the $iVNS$, a traditional VNS, and the $DSOS$, respectively. However, the cBA reported 0.6% (0.24%) superior values compared to our method, while our approach demonstrated a slight advantage of 0.22% when the execution time of our method was adjusted to match the average execution time of cBA ($117s$). The percentages in parentheses denote the performance deviations resulting from an increase in the execution time limit of our method from $40s$ to $60s$.

7. Conclusions

The present study addresses a novel variant of the TSP, known as the Pollution TSP with Refueling (PTSPR). This variant encompasses comprehensive decisions regarding fuel consumption and refueling, which are integrated into the traditional TSP framework. To tackle this new problem, a novel MILP model was developed. Small problem cases were solved utilizing the commercial solver, Gurobi. Moreover, a Two-Stage Double-Adaptive General Variable Neighborhood Search (DA-GVNS) algorithm was developed to effectively solve large-scale instances of the PTSPR. To assess the performance of the proposed solution method, and gain insights into the logistics system's structure and costs, extensive computational experiments and sensitivity analyses were conducted. These analyses examined the impact of potential fluctuations in key model parameters, shedding light on the robustness and adaptability of the proposed approach in addressing real-world logistics challenges.

Based on the extensive analyses conducted in this study, it was deemed necessary to explore additional refueling policies and fuel tank sizes beyond the initially considered options. Our findings highlight the significance of adopting larger fuel tanks in conjunction with more stringent refueling policies, as this combination can yield substantial cost savings. This cost reduction can primarily be attributed to the attainment of a better balance between fuel consumption-related costs and driver wage costs.

The computational and sensitivity analyses conducted in this study have yielded valuable insights, paving the way for several potential avenues of future research. Firstly, there is a need for further exploration and development of alternative refueling policies that strike a harmonious balance between cost considerations and operational efficiency. Moreover, the examination of speed limits and their consequential impact on network structure and refueling requirements is of considerable interest. Additionally, incorporating more realistic scenarios and uncertainties within the model would enhance our understanding of how fluctuations in crucial model parameters, such as fuel prices and demand patterns, impact the structure and cost dynamics of the logistics system. From a technical standpoint, future research efforts could focus on designing new combined local search operators that simultaneously address both routing and refueling decisions, thus enhancing the efficiency of the proposed solution method. Furthermore, considering the adoption of relocation operators instead of swap operators in the post-optimization method holds promise for further improvements.

References

Archetti, C., Peirano, L., & Grazia Speranza, M. (2022). Optimization in multimodal freight transportation problems A Survey. *European Journal of Operational Research*, *299*, 1–20.

Barth, M., & Boriboonsomsin, K. (2009). Energy and emissions impacts of a freeway-based dynamic eco-driving system. *Transportation Research Part D: Transport and Environment*, *14*, 400–410.

Barth, M., Younglove, T., & Scora, G. (2005). *Development of a heavy-duty diesel modal emissions and fuel consumption model*. Technical Report UCB-ITS-PRR-2005-1 California PATH Program, Institute of Transportation Studies, University of California, Berkeley.

Bektaş, T., & Laporte, G. (2011). The Pollution-Routing Problem. *Transportation Research Part B: Methodological*, *45*, 1232–1250.

Bingüler, A., & Bulkan, S. (2015). New variable neighborhood search structure for travelling salesman problems. *British Journal of Mathematics Computer Science*, *6*, 422–434.

Brimberg, J., Salhi, S., Todosijević, R., & Urošević, D. (2023). Variable neighborhood search: The power of change and simplicity. *Computers & Operations Research*, *155*.

Cacchiani, V., Contreras-Bolton, C., Escobar, J., Escobar-Falcon, L., Linfati, R., & Toth, P. (2018). An Iterated Local Search Algorithm for the Pollution Traveling Salesman Problem. In P. Daniele, & L. Scrimali (Eds.), *New Trends in Emerging Complex Real Life Problems* (pp. 83–91). Cham: Springer International Publishing.

Cacchiani, V., Contreras-Bolton, C., Escobar-Falcon, L., & Toth, P. (2023). A matheuristic algorithm for the pollution and energy minimization traveling salesman problems. *International Transactions in Operational Research*, *30*, 655–687.

Cheng, C., Yang, P., Qi, M., & Rousseau, L. (2017). Modeling a green inventory routing problem with a heterogeneous fleet. *Transportation Research Part E: Logistics and Transportation Review*, *97*, 97–112.

Croes, G. A. (1958). A method for solving traveling salesman problems. *Operations Research*, *6*, 791–812.

Demir, E., Bektaş, T., & Laporte, G. (2012). An adaptive large neighborhood search heuristic for the pollution-routing problem. *European Journal of Operational Research*, *223*, 346–359.

Dukkanci, O., Kara, B. Y., & Bektaş, T. (2019). The green location-routing problem. *Computers & Operations Research*, *105*, 187–202.

Ezugwu, A. E., & Adewumi, A. O. (2017). Discrete symbiotic organisms search algorithm for travelling salesman problem. *Expert Systems With Applications*, *87*, 70–78.

Flood, M. (1956). The Traveling-Salesman Problem. *Operations Research*, *4*, 61–75.

Goeke, D., & Schneider, M. (2015). Routing a mixed fleet of electric and conventional vehicles. *European Journal of Operational Research*, *245*, 81–99.

Hansen, P., Mladenović, N., Todosijević, R., & Hanafi, S. (2017). Variable neighborhood search: Basics and variants. *EURO Journal on Computational Optimization*, *5*, 423–454.

Hillier, F. S., & Lieberman, G. J. (2021). *Introduction to Operations Research.* (11th ed.). McGraw Hill.

Hore, S., Chatterjee, A., & Deqanji, A. (2018). Improving variable neighborhood search to solve the traveling salesman problem. *Applied Soft Computing*, *68*, 83–91.

Johnson, D., & Papadimitriou, C. (1985). Performance guarantees for heuristics. In E. Lawler, J. Lenstra, A. Rinnooy Kan, & D. Shmoys (Eds.), *Ch. 5 in The Traveling Salesman Problem.* Chichester: Wiley & Sons.

Karakostas, P., & Sifaleras, A. (2022). A Double-Adaptive General Variable Neighborhood Search algorithm for the solution of the Traveling Salesman Problem. *Applied Soft Computing*, *121*.

Karakostas, P., Sifaleras, A., & Georgiadis, C. (2019a). Basic VNS algorithms for solving the pollution location inventory routing problem. In *Variable Neighborhood Search (ICVNS 2018)* (pp. 64–76). Springer, Cham volume 11328 of *LNCS*.

Karakostas, P., Sifaleras, A., & Georgiadis, C. (2019b). A general variable neighborhood search-based solution approach for the location-inventory-routing problem with distribution outsourcing. *Computers & Chemical Engineering*, *126*, 263–279.

Karakostas, P., Sifaleras, A., & Georgiadis, C. (2020). Adaptive variable neighborhood search solution methods for the fleet size and mix pollution location-inventory-routing problem. *Expert Systems with Applications*, *153*, 113444.

Karakostas, P., Sifaleras, A., & Georgiadis, C. (2022). Variable neighborhood search-based solution methods for the pollution location-inventory-routing problem. *Optimization Letters*, *16*, 211–235.

Khuller, S., Malekian, A., & Mestre, J. (2007). To Fill or Not to Fill: The Gas Station Problem. In L. Arge, M. Hoffmann, & E. Welzl (Eds.), *Algorithms - ESA 2007, LNCS*. Berlin, Heidelberg: Springer volume 4698.

Koç, C., Bektaş, T., Jabali, O., & Laporte, G. (2014). The fleet size and mix pollution-routing problem. *Transportation Research Part B*, *70*, 239–254.

Kowalik, P., Sobecki, G., Bawol, P., & Muzolf, P. (2023). A Flow-Based Formulation of the Travelling Salesman Problem with Penalties on Nodes. *Sustainability*, *15*, 4330.

Menéndez, B., Bustillo, M., Pardo, E. G., & Duarte, A. (2017). General variable neighborhood search for the order batching and sequencing problem. *European Journal of Operational Research*, *263*, 82–93.

Mladenović, N., Todosijević, R., & Urošević, D. (2014). Two level General variable neighborhood search for Attractive traveling salesman problem. *Computers & Operations Research*, *52*, 341–348.

Neves-Moreira, F., Amorim-Lopes, M., & Amorim, P. (2020). The multi-period vehicle routing problem with refueling decisions: Traveling further to decrease fuel cost? *Transportation Research Part E: Logistics* *and Transportation Review*, *133*, 101817.

Or, I. (1976). *Traveling Salesman-Type Combinatorial Problems and Their Relation to the Logistics of Regional Blood Banking*. Ph.D. thesis Department of Industrial Engineering and Management Sciences, Northwestern University Evanston, IL.

Ottoni, A., Nepomuceno, E., Oliveiral, M., & Oliveira, D. (2022). Reinforcement learning for the traveling salesman problem with refueling. *Complex & Intelligent Systems*, *8*, 655–687.

Sahin, M. (2023). Solving TSP by using combinatorial bees algorithm with nearest neighbor method. *Neural Computing and Applications*, *35*, 1863–1879.

Schiffer, M., Schneider, M., & Laporte, G. (2018). Designing sustainable mid-haul logistics networks with intra-route multi-resource facilities. *European Journal of Operational Research*, *265*, 517–532.

Speranza, M. (2018). Trends in transportation and logistics. *European Journal of Operational Research*, *264*, 830–836.

Suzuki, Y. (2012). A decision support system of vehicle routing and refueling for motor carriers with time-sensitive demands. *Decision Support Systems*, *54*, 758–767.

Suzuki, Y. (2014). A variable-reduction technique for the fixed-route vehicle-refueling problem. *Computers & Industrial Engineering*, *67*, 204–215.

Todosijević, R., Mladenović, M., Hanafi, S., Mladenović, N., & Crévits, I. (2016). Adaptive general variable neighborhood search heuristics for solving the unit commitment problem. *International Journal of Electrical Power & Energy Systems*, *78*, 873–883.