jet INTL JOURNAL OF ELECTRONICS AND TELECOMMUNICATIONS, 2026, VOL. 72, NO. 3, PP. 1–9
Manuscript received March 8, 2026; revised July 2026. doi: 10.24425/ijet.2026.157950

Time- and Skill-dependent Vehicle Routing Problem in application of facility maintenance

Radosław Idzikowski, Michał Jaroszczuk and Konrad Pempera

***Abstract*—With current technological advancements, almost all new buildings require periodic maintenance inspections of their installed facilities. These inspections are performed by technicians with varying experience, and therefore changing work hours and rates. However, such inspections often cannot be performed while the building is in use, which requires scheduling visits based on available time windows. In this paper, the described vehicle routing problem with time windows and varying employee rates is formulated using the MILP model. Four solution methods are proposed: a mathematical solver, an adapted Tabu Search (TS) algorithm, an adapted 1nn algorithm, and the Artificial Bee Colony (ABC) algorithm. The purpose of the study is to examine the impact of time window lengths and the junior-senior ratio of the technician groups on the final solution. The results show that for larger instances of the problem, it is not possible to use Gurobi as a solving method due to long computational time. Two implemented metaheuristics, Tabu Search and Artificial Bee Colony algorithm, are comparable with the note that Tabu Search usually achieves better results, depending on the specific spatial characteristics of the instance. The best results were obtained with InsertMove operator for neighborhood exploration for both metaheuristics. In our case, it is proved that the longer tabu list size is applied, the better are the results. Tuning the ABC algorithm showed that it performs best with a number of bees of 75 and a larger limit parameter. Future research may include hybrid use of the proposed metaheuristics and multi-objective optimization to investigate the influence of individual objective function coefficients.**

***Keywords*—Skill-dependent VRP; Building maintenance; Tabu Search; Gurobi; MILP**

I. Introduction

OVER the years, the vehicle routing problem has gained many extensions and formal definitions due to its numerous real-world applications. The basic version of the problem involves finding a distance-optimal set of routes for a fleet of identical vehicles serving a set of customers, with the objective of minimizing the total travel distance or time. Despite the conceptual simplicity of this base model, its extensions and adaptations have proven essential to represent real-life logistics and service-related challenges with more precision. An example would be the incorporation of various operational constraints, such as time windows, heterogeneous fleets, or multi-depot structures, to better reflect practical scenarios. In recent years, increasing attention has been given to hybrid and multi-criteria variants of the vehicle routing problem, which integrate economic, environmental, and human-resource-related aspects into the optimization process [1]. These developments have significantly broadened the applicability of VRP-based models, extending their use beyond traditional logistics to areas such as service management, healthcare logistics, and infrastructure maintenance. Consequently, the flexibility of the VRP formulations enables them to serve as a solid foundation for solving complex resource allocation and scheduling problems in various industrial sectors [2].

One proposed application of the routing problem is to plan tours for crews servicing building automation equipment. Nowadays, virtually every new building must have amenities such as fire protection systems, HVAC (Heating, Ventilation, and Air Conditioning) systems, and personnel movement devices. These installations require regular inspections and maintenance to ensure operational safety and compliance with legal standards. Such inspections are usually performed by mobile technician teams, each with specific qualifications, level of experience, and associated service rates [3]. In addition, each visit requires a certain amount of time depending on the type of repair and the service materials transported by the crews. Planning their work must take into account aspects such as limited operating hours, limited vehicle capacity, and varying rates based on employee qualifications. In turn, planning the order of visiting individual locations must take into account the availability of the location (time window), the location of the building and the required service time. These conditions lead to the formulation of a Heterogeneous Crew Vehicle Routing Problem with Time Windows (HCVRP-TW).

In this paper, a mathematical model for the HCVRP-TW problem is formulated using MILP. The work is an extension of the approach studied in [4] and [5]. Several solving methods are implemented and compared, including a commercial MILP solver, an adapted Tabu Search algorithm, a simple 1-nearest neighbor (1NN) heuristic, and the Artificial Bee Colony algorithm. The main objectives of this study are sixfold:

PG1: Check the status of research on HCVRP-TW problems, especially in application to building maintenance.

Radosław Idzikowski is with Department of Control and Quantum Computing, Wrocław University of Science and Technology, Poland (e-mail: radoslaw.idzikowski@pwr.edu.pl, ORCiD: 0000-0001-5232-1998).

Michał Jaroszczuk is with Department of Control and Quantum Computing, Wrocław University of Science and Technology, Poland (e-mail: michal.jaroszczuk@pwr.edu.pl, ORCiD: 0009-0009-8617-889X).

Konrad Pempera is with Department of Control and Quantum Computing, Wrocław University of Science and Technology, Poland (e-mail: konrad.pempera@pwr.edu.pl, ORCiD: 0009-0002-0701-9639).

PG2: Propose and describe the MILP model for the given HCVRP-TW problem.
PG3: Verify if the problem posed can be efficiently solved using commercial MILP solvers.
PG4: Find hyperparameters for metaheuristic algorithms that will work best for the presented problem instances
PG5: Measure and compare the performance of local search algorithms (Tabu Search) and swarm algorithms (Artificial Bee Colony).
PG6: Find the impact of time windows length and the number of more qualified teams to the problem solution.

The remainder of this paper is organized as follows. Section II provides a detailed literature review of existing approaches to HCVRP-TW and related problems. Section III introduces the problem formulation, model assumptions, and constraints. Section IV presents the computational experiments, followed by the analysis of the results. Finally, Section V summarizes the findings and outlines the directions for future work.

II. Literature

There are many studies about VRP in maintenance technician routing applications. However, to the best of our knowledge, there are only a few works that take into account time windows, differentiation of employee rates depending on skills and shift and the transport of consumables.

Firstly, the works most similar to ours will be presented, containing the VRP extensions we present. One of the most comparable problem considering all extensions introduced in our work is given by Zhao et al. [6] in the application of aircraft maintenance. The problem is solved with CPLEX, EGA-LNS and Genetic Algorithm on real datasets from Beijing Capital International Airport. The results show that it is not possible to compute the problem using the solver due to memory overflow and that the results obtained with EGA-LNS are comparable to GA, but with a longer computation time. Another example of VRP with Time-Windows, Capacity and skills written by Huang et al. [7] consider non-emergency patient transport to hospitals. Unlike us, the authors prohibit the service of certain patients by employees with insufficient skills and consider multi-trip of a vehicle and lunch break. Other paper presented by Yan et al. [8] takes into account the VRP extension with dynamic service times and different rates depending on the skill. To solve the issue, the authors use Dynamic Neighborhood Search algorithm and compare to the results obtained with CPLEX. The experiments carried out took into account a different number of technicians, two objective functions: cost- and time- oriented and the examination of the maximum instance size.

Next, the focus is on works that describe the variable skills and rates of employees, because this is the most valuable feature of this study. In Kiani et al. [9] the authors are modeling VRP with soft time windows and solving them with Genetic Algorithm and Particle Swarm Optimization. What is different to our approach is that the service team is not associated with the vehicle and skills of the technician group does not only affect the cost (wages), but define if a group can service given location. Their conclusion is that in general GA outperforms PSO algorithm, comparing to results obtained with CPLEX. Another example of skill dependent routing problem is given by Schwarze et al. [10] in which, vehicle skills affect the scope of localizations that this vehicle can serve. However, in their model, the costs are skill dependent, but they affect the cost of the route traveled and not the reduction of service time. For the experiments, modified Solomon instances were used, as in our case, and solved with IBM ILOG CPLEX environment. Carried experiments show that in both cases: minimizing total cost and total route length, applying CPLEX solver may improve vehicle load balancing. Similar assumptions are made in Zuo et al. [11] paper in which, however, the problem Solomon's datasets are solved with proprietary heuristics combining Tabu Search and Simulated Annealing algorithms. The conclusions are that the use of multi-skilled employees can reduce the overall cost of enterprise and that a given algorithm is suitable for solving VRP problems with presented extensions.

Finally, papers analyzing VRP problems with time windows in maintenance applications were also reviewed, as some of the concepts described therein may serve as inspiration for this work. A work by Toru and Yilmaz [12] solves Multi-Depot VRP with Time Windows on the example of a Turkish compressor manufacturer. The problem is optimized with Gurobi solver and Clustering Algorithm, which returns worse results, however, with much shorter calculation time. Similarly, Vincezna et al. [13] consider VRPTW in maintenance of energy power plants application. In their model, crew skills are not included, however the vehicle capacity is added as the problem assumes pick-up and delivery of goods. Optimization is performed with Large Neighbourhood Search (LNS) and Simulated Annealing algorithm on real routes. The daily duties of maintenance teams are optimized by Cassettari et al. [14] with multiple criteria using proprietary heuristics. Their model assumes working time limitation, tasks precedence and limited number of vehicles.

All things considered, there are many works analyzing the combination of various employee skills factor with time windows as an extension of VRP. none of them however, to the best of our knowledge, includes the additional delivery of consumables and and shorten service time depending on the skill. The most similar works [6], [7], [8], constitute a good starting point and knowledge base for further considerations. This work seems to fill the gap in research by taking into account additional properties of the problem encountered in practice, such as the mentioned transport of consumables, shortened service time for more experienced employees, shift work and time windows.

III. Problem

This section contains a model of the problem being studied, along with a list of variables used to describe it. The representation used is defined and an example solution is shown. The symbols used in the model are presented in the Table I.

A. Problem assumptions

Before presenting the complete model formulation, some assumptions will be made regarding its properties, presented in the form of key points for clarity. Based on them, a mathematical MILP model of the problem will be presented.

1) There are two levels of qualification for service teams - junior and senior. The number of teams is preset.
2) Two shifts of service crews are introduced. The working time of each shift is half of the depot opening time specified in Solomon's instances.
3) The cost of traveling between two points is independent of the skills of the service team.
4) Each location can be served by a team of any qualification level. If the team has senior qualifications, service time is reduced by 50%, but the cost of the team are twice as high.
5) Overtime of service crews is allowed and paid twice.
6) Exceeding the time window, both from above and below, results in a cost penalty.
7) The labor cost of each team is constant, and applied if the team has at least one location assigned.
8) The service crew carries consumables to perform a service. Vehicle capacity is limited.

B. Definition of variables

In the formulated problem, we assume the existence of r service recipients in the set $\mathcal{R} = \{1, 2, \ldots, r\}$. This set may be extended by location 0, representing the starting node called the depot, creating a set $\mathcal{L} = \{0\} \cup \mathcal{R}$. The depot is a start and end point for each of the service vehicles and has an opening time in the range $[0, d_0]$. Each location $i \in \mathcal{R}$ has defined its coordinates, based on which the travel time between locations is calculated. The travel time is equal to the cost of the route and is represented by the matrix C, where $c_{i,j} \in \mathbb{N}$. Moreover, each location has a defined soft time window $[b_i, d_i] \in \mathbb{N}$, $b_i \leq d_i$, $i \in \mathcal{R}$ and service time $s_i \in \mathbb{N}$. The need to adapt to Solomon's test instances forces the assumption that the service time may be longer than the localization time window. The penalties are constant for each location and given with $e_p, e_q \in \mathbb{N}$. Each location should be visited exactly once and its demand m_i should be delivered to it.

Next, it is necessary to define the set of service crews $\mathcal{V}$. First, the types of crew are established depending on their experience and shift. We assume the existence of morning and afternoon shifts and two experience levels: junior and senior. The working hours of the crew and their costs are given as properties of the crew class as constants $\alpha_v, \beta_v, \gamma_v, \zeta_v$, where $v \in \mathcal{V}$. The depot has the same number of teams with the skills given for each shift. The working time for the morning shift is set in the range $[0, d_0/2]$ and the afternoon shift $[d_0/2, d_0]$. Each vehicle has the same load limit defined by W.

Additionally, we introduce a set of all subtour combinations $S \in \mathcal{S}$ to gain the ability to write a restriction that will ensure that there will be no vehicle routes not connected to the depot. Also, a waiting time $w_{i,v}$ is needed to be set, to model the situation where the team waits idle during the route for the next location's time window to start, to correctly calculate the time of arrival at the location.

TABLE I
NOTATION SUMMARY

Symbol	Meaning
Indexes	
i, j	indexes for locations
k	index for crew type
v	index for crew vehicle
Input data	
$\mathcal{R}, r$	set of service recipients, number of service recipients
$\mathcal{L}$	set of all locations
$\mathcal{V}, v$	set of crews, number of crews
$\mathcal{C}, c_{i,j}$	cost matrix, cost matrix element
b_i	beginning of the location i time window
d_i	end of the location i time window
s_i	service time of location i
m_i	service demand for location i
e_p	penalty for early arrival before location time window
e_q	penalty for late departure after location time window
$\mathcal{V}$, v	set of service crews, number of service crews
α_v	start time of shift for vehicle v
β_v	end time of shift for vehicle v
γ_v	service time multiplier for vehicle v
δ_v	overtime cost multiplier for a vehicle v
ζ_v	basic cost of the shift for vehicle v
W	vehicle load limit
$\mathcal{S}$	a list of subtour combinations
Decision variables	
$x_{i,j,v}$	defines if a vehicle v runs between i and j locations
$y_{i,v}$	defines if a vehicle v is visiting location i
$t_{i,v}$	defines a time of arrival a vehicle v to location i
$p_{i,v}$	defines a time of service before location i time window
$q_{i,v}$	defines a time of service after location i time window
$w_{i,v}$	defines a waiting time before servicing location i
h_v	defines a total working time of vehicle v
a_v	defines calculated overhours of vehicle v

C. Model formulation

The model of a given Heterogeneous Capacitated Vehicle Routing Problem with Time Windows (HCVRP-TW) consists of two parts. As in this paper, the Mixed Integer Linear Programming (MILP) model is used, the first defined element is a goal function $F(\pi) = F_1(\pi) + F_2(\pi) + F_3(\pi) + F_4(\pi)$ that minimizes the total cost of work performed by all teams.

The element $F_1(\pi)$ takes into account the cost of the distance traveled by all vehicles. In the equation, a decision variable x, which defines if a vehicle v runs between the i and j locations, is multiplied by the cost of traveling this arc.

$$F_1(\pi) = \left(\sum_{i \in \mathcal{L}} \sum_{j \in \mathcal{L}} \sum_{v \in \mathcal{V}} x_{i,j,v} c_{i,j} \right) \tag{1}$$

The next factor $F_2(\pi)$ includes into the goal function a cost of overtime hours for each vehicle. This cost is fixed and depends on the level of qualification of the team.

$$F_2(\pi) = \left(\sum_{v \in \mathcal{V}} a_v \delta_v \right) \quad (2)$$

The penalties charged for exceeding time windows when servicing a given location are referred to in $F_3(\pi)$. The variables $p_{i,v}$ and $q_{i,v}$ represent the service time performed before the time window and the service time performed after the time window, respectively. The variables mentioned are multiplied by the penalty factors e_p, e_q for too early arrival and too late departure, respectively.

$$F_3(\pi) = \left(\sum_{i \in \mathcal{R}} \sum_{v \in \mathcal{V}} (e_p p_{i,v} + e_q q_{i,v}) \right) \quad (3)$$

Finally, $F_4(\pi)$, the last part of the goal function, add the basic cost of the crew, if it has at least one location assigned. The assignment of at least one order to a vehicle can be checked based on whether its route has an edge leading to the depot.

$$F_4(\pi) = \left(\sum_{i \in \mathcal{R}} \sum_{v \in \mathcal{V}} x_{i,0,v} \zeta_v \right) \quad (4)$$

The presented goal function is minimized under the following constraints:

$$\begin{aligned}
h_v \geq & \sum_{i \in \mathcal{L}} \sum_{j \in \mathcal{L}} (x_{i,j,v} c_{i,j}) + \\
& + \sum_{i \in \mathcal{R}} (y_{i,v} s_i \gamma_v + w_{i,v}) && \forall v \in \mathcal{V}, && (5) \\
a_v & \geq h_v - (\beta_v - \alpha_v) && \forall v \in \mathcal{V}, && (6) \\
& \sum_{i \in \mathcal{R}} y_{i,v} m_i \leq W && \forall v \in \mathcal{V}, && (7) \\
& \sum_{v \in \mathcal{V}} y_{i,v} = 1 && \forall i \in \mathcal{R}, && (8) \\
y_{i,v} & = \sum_{j \in \mathcal{L}} x_{j,i,v} && \forall i \in \mathcal{R}, v \in \mathcal{V}, && (9) \\
y_{i,v} & = \sum_{j \in \mathcal{L}} x_{0,j,v} && \forall i \in \mathcal{R}, v \in \mathcal{V}, && (10) \\
& \sum_{i \in \mathcal{L}} x_{i,0,v} \leq 1 && \forall v \in \mathcal{V}, && (11) \\
& \sum_{i \in \mathcal{L}} x_{i,i,v} = 0 && \forall v \in \mathcal{V}, && (12) \\
\sum_{j \in \mathcal{L}} x_{i,j,v} & = \sum_{j \in \mathcal{L}} x_{j,i,v} && \forall i \in \mathcal{L}, v \in \mathcal{V}, && (13) \\
\sum_{i \in S} \sum_{j \in S, i \neq j} & x_{i,j,v} \leq |S| - 1 && \forall v \in \mathcal{V}, S \in \mathcal{S}, |S| \geq 2, && (14)
\end{aligned}$$

$$\begin{aligned}
t_{i,v} + c_{i,j} + s_i \gamma_v - M(1 - x_{i,j,v}) & \leq t_{j,v} && \forall v \in \mathcal{V}, i \in \mathcal{L}, j \in \mathcal{R}, && (15) \\
p_{i,v} & \geq b_i - t_{i,v} - M(1 - y_{i,v}) && \forall v \in \mathcal{V}, i \in \mathcal{R}, && (16) \\
q_{i,v} & \geq t_{i,v} + s_i \gamma_v - d_i - M(1 - y_{i,v}) && \forall v \in \mathcal{V}, i \in \mathcal{R}, && (17) \\
w_{i,v} & \geq t_{j,v} - s_i \gamma_v - c_{i,j} \\
& \quad - t_{i,v} - M(1 - y_{i,v}) && \forall v \in \mathcal{V}, i, j \in \mathcal{R}, && (18)
\end{aligned}$$

$$\begin{aligned}
x_{i,j,v} & \in \{0, 1\} && v \in \mathcal{V}, i, j \in \mathcal{L}, && (19) \\
y_{i,v} & \in \{0, 1\} && \forall v \in \mathcal{V}, i, j \in \mathcal{R}, && (20) \\
t_{0,v} & = \alpha_v && \forall v \in \mathcal{V}, && (21) \\
t_{i,v} & \in [\alpha_v, 2d_0] && \forall v \in \mathcal{V}, i \in \mathcal{R}, && (22) \\
p_{i,v}, q_{i,v}, h_v, a_v & \in [0, 2d_0] && \forall v \in \mathcal{V}, i \in \mathcal{R}, && (23) \\
w_{i,v} & \in \{0\} && \forall v \in \mathcal{V}, i \in \mathcal{L} && (24)
\end{aligned}$$

The constraint 5 calculates the total working time of each vehicle, taking into account the travel time, the service time depending on the skills of the crew, and the waiting time between locations. To calculate team overhours, variable a_v which presents the difference between actual and planned working time, is introduced in the constraint 6. As the crews transport the materials to perform the service, it is necessary to check whether their number does not exceed the vehicle's load capacity, which is done in constraint 7. In equation 8 it is written that the location is served by a single vehicle. Setting the state of the variable y to 1 if a vehicle is driving towards it is in turn done in the equations 9 and 10. The constraint 11 ensures that each vehicle that leaves the depot will return to it, and the constraint 12 blocks the creation of tours from the location to itself. The continuity of the tour is guaranteed in 13, which ensures that if the vehicle enters the location, it also leaves it. Preventing subtour creation is done with 14. Calculating the arrival time of vehicle v at a given location is done in 15. The times of exceeding the time windows are calculated in the 16 and 17 equations from above and below, respectively. In constraint 18 waiting time is calculated. Finally the range of decision variables is given in 19, 20, 22, 23 and 24. Additionally, the vehicle's departure time from the depot is set in 21 to ensure that it does not wait at the base after the start of its shift.

IV. Experiment

This section explains the experimental methodology along with a description of the computational environment and datasets used. The research results for each method are then described, along with a commentary on the results obtained.

A. Dataset and implementation description

The experiments were conducted using Solomon benchmark instances [15], which are widely recognized in the vehicle routing problem research community. These instances are divided into six categories based on their characteristics: C1 (clustered locations with narrow time windows), C2 (clustered locations with wide time windows), R1 (randomly distributed locations with narrow time windows), R2 (randomly distributed locations with wide time windows), RC1 (mixed clustered and random distribution of locations with narrow time windows), and RC2 (mixed distribution of locations with wide time windows). Each instance contains 100 customer records with defined coordinates, time window boundaries, and service requirements. The vehicle capacity is proposed by the author for each type of instance category, however for the experiments a constant capacity $W = 100$ was taken.

The problem was adapted to the HCVRP-TW formulation by introducing heterogeneous crew types (junior and senior), dual shift system (morning and afternoon), overtime allowances, soft time windows with penalties, and vehicle capacity constraints for consumables transport. All experiments were executed on a computational platform with following resources: 13th Gen Intel(R) Core(TM) i7-13700K, 32 GB RAM, which ensures consistent performance measurements across all algorithmic approaches. The algorithms and MILP model were written in the C # 12.0 programming language on the .NET 8.0 platform.

B. Algorithms overview and quality measure

For comparative analysis, three algorithmic approaches were implemented: a simple greedy heuristic (1-nearest neighbor), an adapted Tabu Search algorithm, and an Artificial Bee Colony (ABC) algorithm. The greedy heuristic serves as a baseline approach, constructing solutions by iteratively selecting the nearest unvisited location. This kind of method was first presented by Solomon [16], where subsequent locations are chosen based on both distance and time criteria. In addition to route cost minimization, the method reduces penalty components arising from time-window violations or scheduling constraints, while capacity constraints are strictly enforced and therefore cannot be exceeded.

The next algorithm implemented is Tabu Search, the principle of operation of which was presented by Glover in [17]. In this method a list of forbidden movements is applied, to avoid revisiting recent solutions. Thanks to this mechanism, the algorithm can escape local minima, increasing the probability of finding a global one. In the implemented approach, 4 neighborhood search operators were implemented: *insert*, *swap*, *reverse* and *2-opt* to explore the solution space. The performance of each of them is examined in Section IV-C.

The last analysed method is the Artificial Bee Colony (ABC) algorithm proposed by Karaboga [18], which mimics the foraging behavior of honey bees. In the method, three groups of bees are distinguished: workers, onlookers, and scouts. Each worker bee generates a new solution from the neighborhood of the previous solution, which replaces the previous solution if it is better. After the permutation cost calculation process is completed, the worker bees share information about the quality of the verified solutions with the onlooker bees through a dance. The onlooker bees then repeat the process, but they randomly select the previously generated solutions, which means that some solutions will be omitted, while others will be verified multiple times. The better the solution, the greater the probability of its selection. When the no-improvement limit is exceeded, a scout bee is sent to replace the current solution with a random solution.

C. Algorithm comparison on benchmark instances

To compare the quality of given algorithm, we use the Percentage Relative Deviation (PRD) measurement, given in the following formula:

$$PRD(\pi^*) = (REF - F(\pi))/REF \quad (25)$$

In the equation the cost of the solution returned by the examined method $F(\pi)$ is compared to the value of reference solution (REF), which is chosen depending on the type of experiment.

1) Tabu Search parameter sensitivity: The performance of the Tabu Search algorithm was evaluated with different tabu list sizes to determine the optimal configuration. Fig. 1 shows the mean values of the objective function for tabu list sizes of 20, 40, and 80 for all instances. The results indicate mean costs of 11864.21 for size 20, 11832.96 for size 40, and 11514.04 for size 80. The data show a general trend of performance improvement with larger tabu list size, with the most significant reduction occurring when increasing from size 20 to 80, representing a 2.95% improvement.

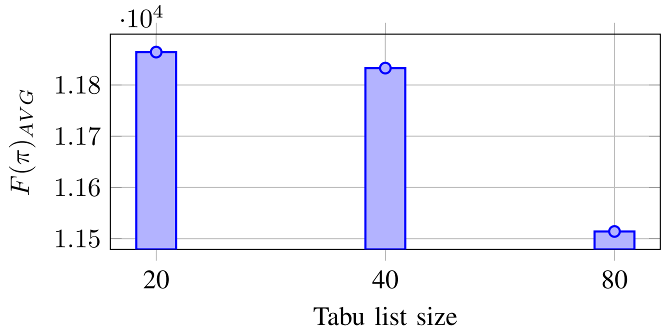

Fig. 1. Average value of the objective function as a function for TS depending on the tabu list size for all instances

In addition, different neighborhood operators were tested for the implementation of Tabu Search. Fig. 2 presents the mean objective values for four types of operators: SwapMove (10430.00), InsertMove (9243.78), ReverseMove (13557.11), and TwoOptMove (13717.39). The InsertMove operator demonstrated the best performance, achieving 11.38% lower costs than SwapMove and 31.95% lower than TwoOptMove. The SwapMove operator was ranked second, while ReverseMove and TwoOptMove showed less favorable results for this variant of the problem.

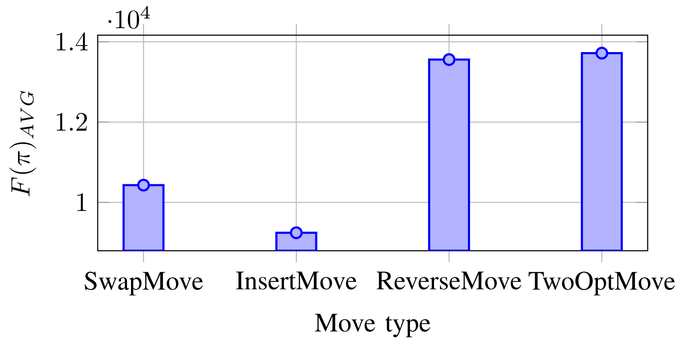

Fig. 2. Average value of the objective function for TS algorithm depending on the move type for all instances

2) Artificial Bee Colony parameter sensitivity: The performance of the Artificial Bee Colony algorithm was analyzed with respect to several key parameters. Fig. 3 shows the impact of colony size on solution quality, testing configurations with 25, 50 and 75 bees. The mean objective values were 20727.27 for 25 bees, 19040.54 for 50 bees, and 18678.06 for 75 bees. The results demonstrate a clear improvement trend with increased colony size, with the 75-bee configuration achieving 9.89% better performance than the 25-bee configuration.

The limit parameter, which controls when scout bees replace abandoned food sources, was tested with values of 20 and 40. Fig. 4 presents mean costs of 20753.13 for limit 20 and 18210.79 for limit 40. The higher limit value resulted in 12.25% better performance, suggesting that allowing more exploration attempts before abandoning a solution region benefits the search process.

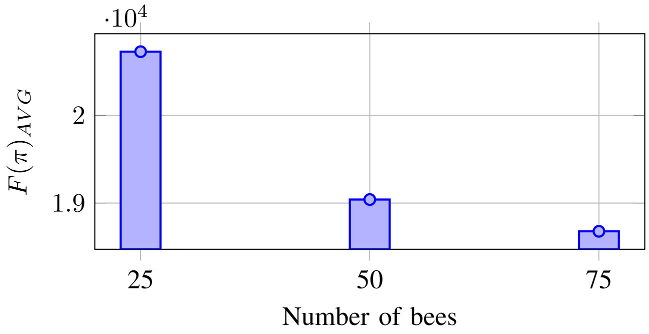

Fig. 3. The average value of the objective function for ABC depending on the number of bees for all instances

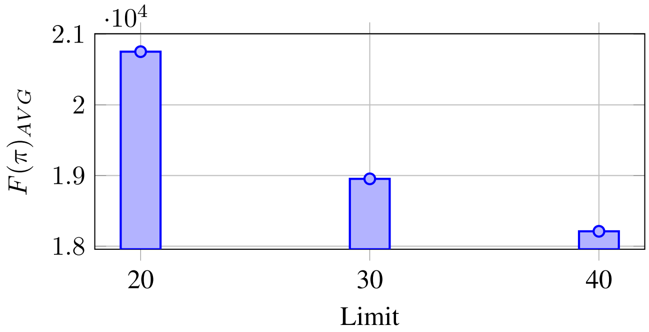

Fig. 4. The average value of the objective function for ABC depending on the limit parameter for all instances

Different types of operator for neighborhood exploration were also evaluated in the ABC algorithm. Fig. 5 shows results for SwapMove (14829.42), InsertMove (13836.47), ReverseMove (21299.36), and TwoOptMove (27962.58). Consistent with Tabu Search findings, InsertMove proved to be the most effective, while TwoOptMove performed the worst, showing an increase of 88.55% in costs compared to InsertMove.

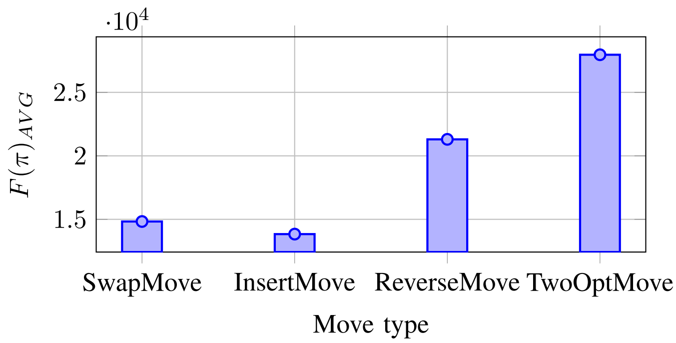

Fig. 5. Average value of the objective function for ABC algorithm depending on the move type for all instances

3) Comprehensive comparison: Firstly, a comprehensive comparison of the three algorithms implemented in all 56 Solomon benchmark instances was made. Across the C-type instances (clustered customers), the Greedy algorithm produced costs ranging from 12625 to 45135. The remaining algorithms performed better, where Tabu Search achieved results between 4523 and 15363, and the ABC algorithm generated costs between 6159 and 20685. For instances of type R (randomly distributed customers), Greedy results ranged from 6621 to 13334, TS from 2347 to 8338, and ABC from 3103 to 9873. RC-type instances (mixed distribution) showed Greedy costs of 8766 to 13946, TS from 2954 to 8047, and ABC from 4320 to 9895.

The improvement ratios confirm that Tabu Search consistently outperformed the Greedy baseline, with improvements ranging from 20.04% to 83.00% in all tested instances. The ABC algorithm also achieved substantial improvements over Greedy, ranging from 6.40% to 79.69%. When comparing TS with ABC, the relative performance varied between instances, with improvements ranging from 2.22% – to 42.79%, indicating that the better method depends on the specific spatial characteristics of the instance.

4) Aggregated results by instance type: To better understand the performance patterns across different problem characteristics, the results were aggregated by instance type. The mean costs for each instance category are presented in Fig. 6. For clustered location instances, the mean costs for C1 were 17201.11 for Greedy, 10233.33 for Tabu Search, and 14239.00 for the ABC algorithm. The C2 instances examination showed mean costs of 31076.88 (Greedy), 7305.75 (TS) and 9524.25 (ABC). For instances with random location positions, R1 gave mean costs of 8564.25 (Greedy), 6419.41 (TS), and 7434.33 (ABC). For R2, the means were 9612.27 (Greedy), 2940.18 (TS), and 4283.63 (ABC). Instances with mixed location types gave means of 9763.63 (Greedy), 7073.5 (TS) and 8429.75 (ABC) for RC1, while RC2 showed means of 11443.88 (Greedy), 3912.12 (TS) and 5634.88 (ABC).

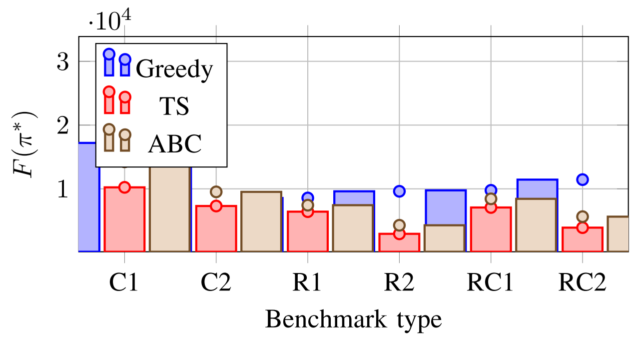

Fig. 6. Comparison of the average values of the algorithm results for each file type

5) Goal function component cost analysis: The total cost function consists of multiple components, each contributing differently to the overall quality of the solution. Table II presents the time window penalty costs across instance types. For C1 instances, the mean penalties were 9566.78 (Greedy), 3456.00 (TS), and 4780.67 (ABC), representing reductions of 63.87% and 50.03%, respectively, over the baseline approach. The C2 instances showed better improvements, with penalties of 24480.00 (Greedy), 1909.00 (TS) and 3859.62 (ABC), corresponding to 92.20% and 84.23% reductions.

The instances of random distribution (R1 and R2) exhibited similar patterns. The penalties for R1 were 1666.83 (Greedy), 950.83 (TS), and 1349.00 (ABC), with reductions of 42.96% and 19.07%. The instances R2 had penalties of 4242.73

(Greedy), 151.82 (TS) and 332.64 (ABC), showing reductions of 96.42% and 92.16%. The mixed distribution instances (RC1 and RC2) demonstrated penalties of 1850.00 (Greedy), 848.50 (TS), 131.50 (ABC) for RC1 and 4881.38, 238.12, 474.88 for RC2, with corresponding reduction percentages.

TABLE II
TABLE COMPARISION OF TOTALPENALTY

File	Greedy	Tabu	Bee	Tabu Greedy	Bee Greedy	Tabu Bee
C1	9566.78	**3456.78**	4780.11	63.87	50.03	27.68
C2	24480.00	**1909.00**	3859.38	92.20	84.23	50.54
R1	1666.83	**913.33**	1181.50	45.21	29.12	22.70
R2	4242.73	**123.73**	358.36	97.08	91.55	65.47
RC1	1850.00	**848.25**	1341.75	54.15	27.47	36.78
RC2	4881.38	**238.88**	474.88	95.11	90.27	49.70

Table III shows the crew deployment costs in algorithms and instance types. For C1 instances, the costs were relatively stable: 3777.78 (Greedy), 3755.89 (TS), and 3711.44 (ABC), showing minimal variation. However, C2 instances demonstrated more significant differences: 3287.50 (Greedy), 1987.00 (TS), and 2100.00 (ABC), representing reductions of 39.54% and 36.12% respectively.

The R1 instances showed crew costs of 3041.67 (Greedy), 2575.00 (TS), and 2683.33 (ABC), with reductions of 15.34 % and 11.78%. R2 instances had costs of 2818.18 (Greedy), 1163.64 (TS), and 1500.00 (ABC), showing reductions of 58.71% and 46.77%. The RC-type instances followed similar trends, with RC1 showing costs of 3337.50 (Greedy), 2775.50 (TS), 2975.00 (ABC) and RC2 showing 3212.50 (Greedy), 1525.00 (TS), 1937.50 (ABC).

TABLE III
TABLE COMPARISION OF CREWCOST

File	Greedy	Tabu	Bee	Tabu Greedy	Bee Greedy	Tabu Bee
C1	3777.78	3755.56	**3711.11**	0.59	1.76	-1.20
C2	3287.50	**1987.50**	2100.00	39.54	36.12	5.36
R1	3041.67	**2575.00**	2683.33	15.34	11.78	4.04
R2	2818.18	**1163.64**	1500.00	58.71	46.77	22.42
RC1	3337.50	**2775.00**	2975.00	16.85	10.86	6.72
RC2	3212.50	**1525.00**	1937.50	52.53	39.69	21.29

6) Gurobi solver scalability analysis: To establish a baseline for the computational complexity of the problem and determine the maximum instance size solvable by exact methods within reasonable time limits, a scalability study was conducted using the Gurobi optimization solver. The study evaluated the solver's performance on instances of varying sizes, ranging from 9 to 15 customers, with a time limit of 10 minutes (600 seconds) per instance. Six representative problem types were tested.

The results, presented in Fig. 7, demonstrate an exponential growth in the solution time as the size of the instance increases. For instances with 9 customers, the mean solution time was approximately 1.7 seconds, indicating that very small instances can be solved almost instantaneously. As the problem size increased to 10 and 11 customers, the mean solution times increased to 4.9 and 10.7 seconds, respectively, showing the first signs of the growth of computational complexity.

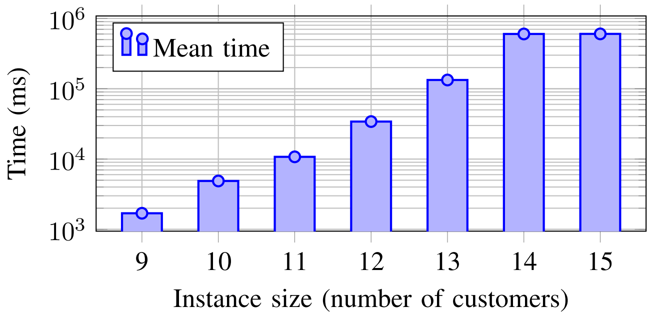

Fig. 7. Mean solution time for Gurobi solver by instance size (10 minute time limit)

A significant jump in computational requirements occurred at 12 customers, where the mean solution time increased to approximately 34 seconds, more than three times the time required for 11-customer instances. The complexity continued to grow dramatically for 13-customer instances, requiring an average of 133 seconds (approximately 2.2 minutes) to reach optimality. This represents nearly a four-fold increase compared to the 12-customer instances.

The critical threshold was reached at 14 customers, where all tested instances exceeded the 10 minute time limit, with solution times reaching approximately 601 seconds (10 minutes). Similarly, instances with 15 customers also exceeded the time limit. These results indicate that the maximum reliably solvable instance size using Gurobi within a 10 minute time constraint is 13 customers. This proves that the commercial MILP solver Gurobi cannot be used in practical applications and on real datasets due to its too long computation time.

7) Comparison with Gurobi optimal solutions: To further assess the effectiveness of the metaheuristics implemented, the results obtained with them were compared with the exact solutions returned by the Gurobi solver. This comparison was conducted on all Solomon instances for which the exact solver could compute provably optimal solutions within the 10-minute time limit. As discussed in Section IV-C6, this corresponds to instances containing up to approx. 13 customers.

Table IV presents a subset of representative results comparing Gurobi with Tabu Search and Artificial Bee Colony metaheuristics. The full dataset includes 56 instances, but only a short illustrative excerpt is shown here for clarity.

The results confirm that neither Tabu Search nor the ABC algorithm surpassed the optimal solutions provided by Gurobi on any of the instances tested, which is consistent with expectations for exact vs. heuristic methods. However, a noteworthy observation is that in several cases both heuristics matched the optimal solution exactly (instances marked with bold in Table IV), demonstrating that the implemented metaheuristics are capable of reaching optimal-quality solutions on small problem sizes.

Across the entire dataset, the deviation from the optimal solution remained small. Tabu Search consistently produced

TABLE IV
SAMPLE COMPARISON OF METAHEURISTIC RESULTS WITH GUROBI OPTIMAL SOLUTIONS

Instance	Tabu Search	ABC	Gurobi (Opt.)
C102	**789**	793	789
C104	**713**	716	713
C106	1456	1456	1302
C108	950	950	801
C202	1525	1525	1370
C206	2201	2201	1739
C208	1707	1707	1260
R102	**890**	**890**	890
R104	**784**	**784**	784
R106	**830**	**830**	830
R108	**748**	**748**	748
R202	905	799	724
R204	801	747	664
R206	689	635	562
R208	622	585	505
RC102	**1001**	1007	1001
RC104	977	981	975
RC106	**1054**	1063	1054
RC108	994	1008	994
RC202	**1171**	**1171**	1171
RC204	**1127**	**1127**	1127
RC206	**1186**	1200	1186
RC208	**949**	951	949

results very close to optimal values, reflecting its strong intensification capabilities. The ABC algorithm also performed competitively, often approaching the optimal solution and occasionally matching it.

These findings underscore an important distinction: while exact solvers guarantee globally optimal solutions, their scalability is severely limited, making them impractical for large-scale VRP instances. Metaheuristics, while not guaranteeing optimality, maintain high-quality performance across all instance sizes evaluated in this study and remain computationally feasible for real-world problem dimensions.

D. Findings and interpretation

The experimental results reveal several important patterns regarding the performance of different algorithmic approaches for the HCVRP-TW problem. First, both metaheuristic approaches (Tabu Search and Artificial Bee Colony) substantially outperform the simple greedy baseline across all instance types, demonstrating the value of sophisticated search strategies for this complex optimization problem. The magnitude of improvement varies significantly with instance characteristics, with the highest gains observed in C2 and R2 instances (those with wide time windows).

The superior performance on wide time window instances can be attributed to the larger solution space these problems afford. Wide time windows provide more flexibility in scheduling visits, allowing metaheuristics to explore diverse routing configurations and discover solutions that effectively balance multiple cost components. Conversely, narrow time window instances (C1, R1, RC1) impose stricter temporal constraints that limit solution space exploration, resulting in smaller but still substantial improvements over the greedy approach.

Comparing the two metaheuristics, Tabu Search generally achieves lower total costs than the ABC algorithm across most instances. However, the ABC algorithm demonstrates competitive performance and, in some cases, matches or exceeds Tabu Search results, particularly on certain R-type and RC-type instances. This suggests that both algorithms have merit and that their relative effectiveness depends on the characteristics of the specific problem.

The cost component analysis provides information on how the algorithms achieve their improvements. Substantial reductions in time window penalties indicate that both metaheuristics excel at finding routes that respect customer availability constraints. This is particularly evident in the C2 and R2 instance categories, where penalty reductions exceed 90% compared to the greedy baseline. Crew cost reductions demonstrate that metaheuristics successfully consolidate customer visits onto fewer routes, thus minimizing the number of deployed crews and associated fixed costs.

Parameter sensitivity analysis reveal important tuning considerations. For Tabu Search, larger tabu list sizes (80) provide better performance than smaller sizes (20), suggesting that maintaining a more extensive memory of recent moves helps avoid local optima. The InsertMove operator proves to be most effective for neighborhood exploration in this problem domain, likely because insertion operations provide a good balance between solution modification intensity and efficiency.

For the Artificial Bee Colony algorithm, increasing colony size from 25 to 75 bees improves performance, though with diminishing returns. The limit parameter analysis indicates that allowing more exploration attempts before abandoning solution regions (limit=40 vs limit=20) benefits solution quality. The consistency in operator performance between TS and ABC - with InsertMove outperforming other operators in both cases - suggests that this operator type is fundamentally well-suited to the neighborhood structure of HCVRP-TW problems.

A comparison with optimal solutions obtained using Gurobi provides additional insight. For all instances solvable within the 10-minute time limit, the metaheuristics produced solutions very close to the optimal ones, confirming their high effectiveness on small-scale problems. In several cases (e.g., R102–R105 and RC101–RC103) TS and in some instances ABC, reproduced the optimal solution exactly. This consistency with the solver demonstrates that both methods are capable of reaching optimal-quality solutions whenever the structure and size of the instance make such outcomes attainable.

The Gurobi scalability study further contextualises these results. While the solver efficiently handles instances with up to 13 customers, computation times increase exponentially, and instances with 14 or more customers cannot be solved within the acceptable time. This confirms that exact methods are impractical for realistic problem sizes. In contrast, both Tabu Search and ABC scale effectively to all 56 benchmark instances while maintaining high-quality performance, making them suitable for real-world applications where problem sizes exceed the reach of exact optimisation.

V. CONCLUSIONS

The computational experiments demonstrate that the HCVRP-TW problem, incorporating heterogeneous crews, dual shifts, overtime considerations, and soft time windows, can be effectively addressed using metaheuristic optimization approaches. The implemented Tabu Search and Artificial Bee Colony metaheuristic algorithms both produce high-quality solutions that substantially improve upon baseline greedy construction heuristics, with limited computational time.

Concerning PG6 (impact of time windows and crew qualifications), the aggregated results by instance type clearly demonstrate that the width of the time window significantly affects the difficulty of the solution and the magnitude of achievable improvements. Instances with wide time windows (C2, R2, RC2) show the largest improvements from metaheuristic optimization, while narrow window instances exhibit more modest but still meaningful gains. Crew cost analysis reveals that algorithm sophistication enables better crew utilization and route consolidation.

In addition, the comparison with optimal solutions returned by Gurobi solver, confirms that the metaheuristics consistently produce solutions close to the global optimum on instances of tractable size. The scalability limitations of exact methods further highlight the practical value of heuristics, as the problem becomes computationally intractable even for moderate instance sizes, whereas Tabu Search and ABC maintain both robustness and efficiency across all tested cases.

Future research directions suggested by these results include: (1) hybrid approaches combining Tabu Search and Artificial Bee Colony to leverage their complementary strengths, (2) adaptive parameter tuning mechanisms that adjust algorithm behavior based on instance characteristics, (3) multi-objective optimization variants that explicitly trade off different cost components, and (4) investigation of parallel computing strategies to further improve solution quality within practical time limits.

The experimental evidence supports the conclusion that metaheuristic approaches provide an effective and practical solution methodology for real-world building maintenance routing problems with complex constraints. The insights gained regarding parameter sensitivity and algorithm behavior provide guidance for practitioners implementing these methods in operational settings.

REFERENCES

[1] T. Vidal, T. G. Crainic, M. Gendreau, and C. Prins, "Heuristics for multi-attribute vehicle routing problems: A survey and synthesis," *European Journal of Operational Research*, vol. 231, no. 1, pp. 1–21, 2013. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0377221713002026

[2] V. Pillac, M. Gendreau, C. Guéret, and A. L. Medaglia, "A review of dynamic vehicle routing problems," *European Journal of Operational Research*, vol. 225, no. 1, pp. 1–11, 2013. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0377221712006388

[3] F. W. Yik and J. H. Lai, "The trend of outsourcing for building services operation and maintenance in hong kong," *Facilities*, vol. 23, pp. 63–72, 2005.

[4] K. Pempera and M. Jaroszczuk, "Analysis of the impact of time window lengths on maintenance vehicle routing problem efficiency," *Advances in Dependable Systems and Networks : Proceedings of the Twentieth International Conference on Dependability of Computer Systems DepCoS-RELCOMEX*, vol. 1427, pp. 181–190, 2025.

[5] R. Idzikowski, M. Jaroszczuk, P. Nowak, and M. Uchroński, "Tabu search algorithm for cost optimization in multimodal transport," *IEEE 24th International Symposium on Computational Intelligence and Informatics (CINTI)*, pp. 93–97, 2024.

[6] P. X. Zhao, M. N. Dai, X. Han, C. Xu, and C. C. Du, "Model and algorithm for the skill capacitated vrp with time windows in airports," *International Journal of Simulation Modelling*, vol. 22, pp. 133–144, 2023.

[7] N. Huang, H. Qin, Y. Du, and L. Wang, "An exact algorithm for the multi-trip vehicle routing problem with time windows and multi-skilled manpower," *European Journal of Operational Research*, vol. 319, pp. 31–49, 2024.

[8] X. Yan, B. Xiao, Y. Xiao, Z. Zhao, L. Ma, and N. Wang, "Skill vehicle routing problem with time windows considering dynamic service times and time-skill-dependent costs," *IEEE Access*, vol. 7, pp. 77 208–77 221, 2019.

[9] M. Kiani, H. Seidgar, and I. Mahdavi, "Scheduling multi-skilled manpower with considering teams replacement and site-dependent vehicles routing," *Int. J. Mathematics in Operational Research*, vol. 10, pp. 49–68, 2016.

[10] S. Schwarze and S. Voß, "Improved load balancing and resource utilization for the skill vehicle routing problem," *Optimization Letters*, vol. 7, pp. 1805–1823, 2012.

[11] Z. Zuo, Y. Li, J. Fu, and J. Wu, "Human resource scheduling model and algorithm with time windows and multi-skill constraints," *Mathematics*, vol. 7, 2019.

[12] E. Toru and G. Yilmaz, "A multi-depot vehicle routing problem with time windows for daily planned maintenance and repair service planning," *Pamukkale University Journal of Engineering Sciences*, vol. 29, p. 913–919, 2023.

[13] C. Vincenza, D. D. Francesco, A. Longheu, M. Malgeri, G. Mangioni, A. Romeo, and N. Trapani, "Pick-up & deliver in maintenance management of renewable energy power plants," *Proceedings of the Federated Conference on Computer Science and Information Systems*, vol. 21, pp. 579–585, 2020.

[14] L. Cassettari, M. Gaggero, and S. Saccaro, "Optimization of gas metering maintenance services: A multiobjective vehicle routing problem with a set of predefined overlapping time windows," *Networks*, vol. 78, pp. 284–302, 2021.

[15] M. M. Solomon, "Solomon vrptw benchmark," 2021. [Online]. Available: https://www.kaggle.com/dsv/2355681

[16] M. Solomon, "Algorithms for the vehicle routing and scheduling problems with time window constraints," *Operations Research*, vol. 35, pp. 254–265, 1987.

[17] F. Glover, "Tabu search—part i," *ORSA Journal of Computing*, vol. 1, 1989.

[18] D. Karaboga, "An idea based on honey bee swarm for numerical optimization," 2005.