Acta logistica - International Scientific Journal about Logistics
Volume: 12 2025 Issue: 2 Pages: 253-260 ISSN 1339-5629

https://doi.org/10.22306/al.v12i2.600 *Received: 31 July 2024; Revised: 27 Mar. 2025; Accepted: 06 May 2025*

Postal optimization by three metaheuristics – a case study

Grzegorz Koloch
Warsaw School of Economics, Al. Niepodległości 164, 02554, Warsaw, Poland, EU,
gkoloch@sgh.waw.pl

Keywords: logistics, optimization, real-life postal delivery problem, metaheuristics.

Abstract: The efficiency of postal delivery services impacts various aspects of business operations as well as the daily lives of individuals. With the surge in online shopping, increasing expectations for timely deliveries, and intense competition, postal operators are under pressure to optimize their transportation networks. This poses a significant challenge for traditional postal systems, particularly in such areas as route optimization, resource allocation and network planning. Postal operators recognize that optimizing transportation routes is a critical task to ensure cost-effectiveness and customer satisfaction, which directly influences their business performance and market share. In this paper, we analyze three possible approaches to solving a real-life, practical instance of a postal transportation plan optimization problem. Specifically, we evaluate the performance of three metaheuristic methods: Simulated Annealing, Tabu Search, and a Genetic Algorithm. We analyze which approach performs best in a real-life scenario inspired by the operations of one of the biggest postal operators in Central and Eastern Europe. This scenario mixes elements of multiple standard routing problem specifications, like capacity constraints of vehicles and network nodes, time windows, pickups and deliveries or multiple types of vehicles.

1 Introduction

Optimization of logistics and transportation networks is a key challenge in many industries, particularly in sectors where cost efficiency, timely deliveries, and resource management are critical. Postal service operations, which involve the movement of large volumes of mail and parcels across extensive networks, require advanced optimization techniques to improve performance while keeping costs under control. Given the combinatorial complexity of such problems, traditional optimization methods often struggle to find high-quality solutions for problems that involve practical constraints. As a result, metaheuristic algorithms have gained popularity as effective approaches for tackling large-scale, complex transportation optimization problems [1].

This study examines the application of metaheuristic optimization techniques to a real-world postal service problem, evaluating their ability to enhance transportation planning and cost efficiency. Specifically, we compare the performance of three popular algorithms – Simulated Annealing, Tabu Search, and a Genetic Algorithm, in improving pre-optimized transportation plans within a limited computational budget. These algorithms, known for their effectiveness in combinatorial optimization, are tested under real-world constraints to determine their suitability for postal logistics planning. The case study presented in this paper incorporates multiple elements into a single problem specification, including time windows, vehicle capacities, postal outlets capacities, operating costs of vehicles and postal outlets, multiple vehicle types, multiple network connection types (standard and express), compatibilities between vehicles and postal outlets or varying demand densities (low, expected, and peak loads). Therefore, contribution of this paper is twofold. First, it provides a description of a real-life postal optimization problem faced by one of the largest postal operators in Central and Eastern Europe, with the goal of enriching the understanding of the complexities involved in practical optimization of transportation problems encountered by such companies. Second, it presents a comparative study of three optimization algorithms. These methods are applied to the same postal transportation case study, and their results are reported and compared. As a consequence, the paper seeks to offer insights into the applicability and effectiveness of the tree algorithms. We focus here on an operational level optimization of postal delivery plans, i.e. we assume that the structure of the transportation network is fixed. Scenarios involving problem instances, where technical parameters of the network elements can be adjusted by the algorithm, i.e. a brownfield scenario, and where only the Genetic Algorithm was used, were analyzed in [2], where the aim was to identify network elements that could serve as hubs for satellite locations.

The remainder of this paper is organized as follows. Section 2 provides a brief review of the main types of transportation problems and of the three metaheuristic algorithms we employed for optimization. Section 3 presents the assumptions of the case study that we consider as a reference scenario for testing the algorithms. Section 4 details the methodological aspects of the tested algorithms. This is followed by a discussion of the results, presented in Section 5, where results of numerical experiments are reviewed, and the algorithms are compared in terms of the quality of obtained solutions and the robustness of their performance. Finally, Section 6 concludes.

2 Literature review

Postal transportation systems play a crucial role in facilitating global commerce, trade, and communication, making them essential for the efficient functioning of modern societies [3]. With the rapid growth of e-commerce and the increasing demand for quick and reliable delivery services, optimizing postal transportation plans has become a critical task for postal operators [4]. To ensure timely delivery of mail and packages, postal service providers are increasingly focused on enhancing the efficiency of their networks and of their transportation plans [5]. As a consequence, over the years, researchers and practitioners have focused on developing and testing approaches to enhance the performance of transportation operations [6-8].

Technically, the problem analyzed in this article falls into the category of Vehicle Routing Problems (VRP), which serves as a quintessential model for addressing logistical challenges, including elements such as optimization of delivery routes, vehicle assignments, and resource utilization [9,10]. In essence, VRPs address the question of how to best allocate vehicles – and potentially drivers – to deliver cargo (e.g., mail or packages) or to provide services at specified locations. VRPs come in various forms, each presenting unique challenges and complexities. Among these, several types of VRP can be distinguished [11], for example: Capacitated VRP (CVRP), where vehicles have limited carrying capacities, and the goal is to meet customer demands while adhering to these capacity constraints, Heterogeneous Vehicle Routing Problem (HVRP), involving a fleet of vehicles with diverse characteristics, such as capacity, speed, cost, fuel usage, or the type of cargo they can carry, VRP with Time Windows (VRPTW), where customers must be served within specific time slots, introducing a temporal dimension to route planning, Multi-Depot VRP (MDVRP), where vehicles can be dispatched from multiple depots, adding optimization opportunities, Pickup and Delivery VRP (PDVRP), where vehicles perform both pickup and delivery operations, with each pickup task paired with a corresponding delivery task, and routes requiring adherence to specified sequences of pickups and deliveries, Split Delivery Vehicle Routing Problem (SDVRP), which relaxes the constraint that each customer must be visited exactly once, allowing split deliveries, Vehicle Routing Problem with Backhauls, where all deliveries on a route must be completed before any pickups are made, and Dynamic VRP (DVRP), where problem parameters, such as customer demands or vehicle availability, change over time, requiring real-time adaptation of routes. There are also other variants of VRPs, however, practical scenarios typically are more complex than theoretical models as they mix different variants of VRPs. The version of a VRP considered in this article falls into this category and it can be classified as a Rich VRP [12]. This means that it combines various elements of such variants as CVRP, VRPTW and others.

In this paper we aim to evaluate the suitability for application of metaheuristic optimization algorithms to a practical instance of a Rich VRP experienced by a postal service provider. We use three popular algorithms – Simulated Annealing (SA), Tabu Search (TS), and the Genetic Algorithm (GA). These algorithms were selected for benchmarking and comparison as each of them follows a distinct optimization strategy: SA employs a probabilistic search process that balances exploration and exploitation, TS utilizes a structured, memory-based approach to refine solutions, and GA applies evolutionary principles to iteratively enhance quality of solution. The SA algorithm was introduced by Kirkpatrick et al. in 1983 [13] and has since become a widely used optimization technique for solving complex optimization problems, both continuous and discrete ones. It is inspired by the physical process of annealing used to determine low energy states of physical systems. At the beginning of the search space exploration, it acts more like a random search method, and it gradually shifts towards a greedy search state. This state transition is governed by a parameter called the temperature. The TS method was introduced by Glover in the late 1980s [14] and it can be perceived as an enhancement of SA, where the search space exploration process is more guided, i.e. the neighborhood of the current solution is more intensively explored before a transition to the new solution is performed. To avoid a local minimum trap, the algorithm uses a mechanism called a tabu list, which prevents it from cycling around local minima. GA was first introduced earlier, by Holland, in the 1960s [15], and also has become a widely recognized optimization method across various fields. It is based on the principle, where new solutions are constructed from the existing ones, with elements of randomness involved. By employing numerical procedures called genetic operators – selection, reproduction and mutation, it intends to mimic the process of natural selection and evolution.

3 Problem formulation

The problem considered in this paper falls into the category of so-called rich or practical VRPs. These problems aim to determine optimal routes for a fleet of vehicles while accounting for a variety of practically important constraints. In particular, the postal delivery version of the VRP analyzed in this paper is based on the following assumptions.

- Each node in the network represents a postal operations outlet. The network comprises 14 main nodes that serve as major logistics platforms, 12 local or regional network nodes, 180 distribution and reloading nodes, approximately 8,000 postal service nodes, and about 34,000 post boxes. Postal service nodes and postal outlets are integral to the first- and last-mile optimization sub-problem.
- Outlets operate within specific hours (some 24/7, others not) and have diverse technical characteristics

depending on their size, location, construction, and installed equipment. Each outlet has a defined list of vehicle types it can accommodate, along with its capacity in terms of the number of vehicles it can process per hour. For each cargo type, an outlet is characterized by the volume it can handle per hour, the unit cost of processing, its storage capacity, and the unit storage cost. These volumes, times, and costs primarily depend on the size of the outlet and the types of installed equipment.

- Several types of vehicles with capacitated storage spaces are available, as in the Capacitated Vehicle Routing Problem, ranging from smaller delivery cars to various sizes of vans and larger trucks. Each vehicle type has a defined capacity for different types of standardized cargo (e.g., boxes and pallets) and an average traveling speed, which depends on the time of day and the chosen route. Additionally, there are special-purpose vehicles designed to carry specific types of cargo. Each vehicle type also has unique loading and unloading times. Vehicles incur costs per hour and per kilometer, while drivers have defined hourly rates. These characteristics classify the problem as a variant of the heterogeneous fleet VRP. Since the postal operator outsources its transportation services, the costs are determined by an external fleet service provider responsible for delivering the required number of vehicles of various types to the respective locations. Consequently, we assume the number of vehicles of each type is unlimited.
- Vehicles operate from multiple depot locations, as in the Multi-Depot Vehicle Routing Problem. They can begin and end their routes at any of the locations that function as depots or destination points. All network nodes can serve as both pickup and delivery sites. The first and last miles are pre-optimized and integrated into the problem as parameters that specify the time required to manage pickups and deliveries during the first and last mile.
- When delivering cargo to postal outlets, a vehicle arriving at a network node that is currently busy serving other vehicles must queue and wait according to the first-in, first-out principle.
- Cargo is shipped in standardized forms, such as boxes and pallets. Mail is transported in boxes, while parcels are carried on pallets, both of which must be picked up from and delivered to designated locations – nodes of the transportation network – as in the Pickup and Delivery Vehicle Routing Problem. In practice, cargo shipment demands exist between all pairs of locations.
- Vehicles travel along routes connecting network nodes, with travel times between nodes varying throughout the day. There are two types of network connections: standard and express. In the transportation plan, when a vehicle travels between two locations, the type of network connection must be specified. Typically, optimized solutions assign trucks and vans to standard network connections, while smaller, more flexible vehicles utilize express connections.
- Each postal outlet must be served within defined time windows, as in the Vehicle Routing Problem with Time Windows. Some outlets operate with longer time windows to accept pickups and deliveries, while others have more restrictive time constraints. To ensure cargo is delivered to its final destination on a given day, it must reach the designated outlet before a specified cutoff time, usually during the night.

An important aspect of the problem formulation are business objectives. In these terms, it is assumed that the operator aims to minimize operational costs, including penalties for violating Service Level Agreement (SLA) constraints. SLA constraints vary depending on the type of cargo being shipped. For mail, SLA ranges from next-day delivery for priority shipments to delivery within three days for economy shipments. The same timeframes apply to parcels. However, pallets must be delivered under a next-day regime.

Standard versions of VRP are challenging itself. The number of possible routes grows exponentially with the number of network locations and vehicles, leading to a vast solution space that is computationally expensive to explore. This phenomenon is known as combinatorial explosion. The practical version of the problem that we consider here, involves numerous simultaneous constraints, such as vehicle capacity and time windows, which further complicate the problem and narrow the set of feasible solutions. Optimizing such practical VRPs requires carefully balancing trade-offs between conflicting objectives, such as minimizing travel costs or distances while maximizing vehicle utilization. To tackle such complexities, numerous methods have been explored over the years, including exact algorithms, metaheuristics, and hybrid approaches. Exact techniques, such as dynamic programming and mixed-integer programming, employ methods like branch-and-bound, cutting plane, and column generation to iteratively narrow the search space and explore selected subregions in pursuit of an optimal solution. However, these methods are most effective when the problem formulation does not involve nonlinear interactions between problem elements. They also often struggle with scalability. As a result, exact methods excel when applied to more stylized VRP instances, particularly when the problem can be naturally formulated as a mixed-integer linear program. Metaheuristic algorithms, such as Genetic Algorithms, Tabu Search, Simulated Annealing, Particle Swarm Optimization or Ant Colony Optimization, provide, on the other hand, an efficient way to explore the search space in cases where problem formulations deviate from simplified scenarios. These methods iteratively refine solutions using heuristic guidance, enabling them to find high-quality, often near-optimal, solutions within a reasonable computational time. This makes them wellsuited for real-world applications. Hybrid methods combine the strengths of different approaches by integrating exact and metaheuristic techniques to enhance performance and robustness compared to standard metaheuristics. In the following sections, we test three popular metaheuristic approaches. In what follows we consider the metaheuristic approach to solving the case study.

4 Methodology

Although optimizing logistics and transportation processes is essential for improving resource utilization and enhancing overall operational efficiency, the combinatorial nature and complexity of these problems pose significant challenges for traditional optimization methods. Over the years, metaheuristic algorithms have emerged as powerful tools for addressing these challenges, providing effective and efficient solutions to a wide range of optimization problems in logistics and transportation. Metaheuristic algorithms are iterative optimization techniques capable of exploring vast solution spaces. While they do not guarantee finding an optimal solution, their goal is to identify high-quality solutions within a reasonable computational time. These algorithms often draw inspiration from natural processes, social behaviour, and mathematical principles to guide the search process [16]. Some of the most widely used metaheuristic algorithms for optimization in logistics and transportation include:

- Genetic Algorithms (GA) that are inspired by the process of natural selection and evolution. They operate on a population of candidate solutions, iteratively evolving them through selection, reproduction, and mutation, which, as numerical procedures, are referred to as genetic operators.
- Simulated Annealing (SA) is based on the concept of sampling the next solution from the neighborhood of the current one, always accepting moves to better solutions while occasionally accepting moves to worse solutions with a controlled probability. This approach is inspired by the annealing process in metallurgy, where a material is heated and then gradually cooled to reach a low-energy state.
- Tabu Search (TS) is an optimization algorithm that explores the solution space by iteratively transitioning from one solution to a neighboring one while avoiding previously visited regions. It incorporates a memory mechanism to guide the search and prevent cycling back to recently explored search space regions.
- Particle Swarm Optimization (PSO) is inspired by the behavior of bird flocks and fish schools. It maintains a swarm of solutions (particles) that iteratively adjust their positions based on both individual and collective experience of particles, guiding the search toward promising search space regions.
- Ant Colony Optimization (ACO) mimics the foraging behavior of ants, where solutions are constructed iteratively by simulating the pheromone trails that ants lay down to communicate and reinforce optimal paths. This mechanism guides the search process toward more promising solutions over time.

Some other notable examples of metaheuristics include Iterated Local Search and Variable Neighbourhood Search. In general, metaheuristic algorithms are powerful tools for tackling complex optimization problems within a reasonable timeframe. By drawing inspiration from natural and social phenomena, they can generate effective and efficient sub-optimal solutions for a wide range of problems. In the remainder of this chapter, we provide a brief overview of the metaheuristic methods used in this study.

4.1 *Simulated annealing*

It is highly versatile and applicable across various practical domains, including the Traveling Salesman Problem, Vehicle Routing Problem, job scheduling, manufacturing, asset allocation problems, and the estimation of meta-parameters in statistical and machine learning models. SA avoids getting trapped in local extrema and seeks to converge toward a global optimum by occasionally accepting steps that temporarily worsen the objective function's value. The algorithm is relatively simple to implement and requires minimal problem-specific parameter tuning. The initialization phase involves generating an initial solution x_0, setting the initial value of the temperature parameter $t_0 > 0$, and setting the annealing schedule parameter α, which governs how temperature drops from one iteration to the other. The initial solution is randomly sampled from the search space and pre-optimized using simple heuristics, such as a push-forward insertion heuristic. The initial temperature t_0 is calibrated through a trial-and-error process, while the temperature update parameter α is set to ensure that the algorithm converges when the computational budget is exhausted. At each iteration, a candidate solution x'_k is drawn uniformly from the neighborhood of the current solution $N(x_k)$. The neighborhood consists of solutions that can be obtained from x_k by applying small random perturbations. These perturbations modify certain elements of the solution x_k, such as the grouping of cargo bundles, the assignment of vehicles to routes, or the order in which network nodes are visited. As the activation function, we use the Metropolis function, given by (1).

$$p = \min\left(1, \exp\left(-\frac{f(x'_k)-f(x_k)}{t_k}\right)\right) \quad (1)$$

As the temperature update schedule, which gradually decreases the temperature over iterations, we use an exponential decay schedule defined as $t_{k+1} = \alpha \times t_k$. The temperature decay parameter α is calibrated so that after all

K iterations, the final temperature is $t_K = 10^{-2}$. The parameter K is set to fit the available computational time budget.

4.2 Tabu search

In each iteration, it performs an intensive Monte Carlo exploration of the neighborhood $N(x_k)$ of the current solution x_k and selects the move to the next solution x_{k+1} that results in the greatest improvement to the objective function. Solutions within the neighborhood $N(x_k)$ are generated by perturbing the current solution x_k, however, according to the tabu list principle, perturbations applied within the last n iterations cannot be undone, see (2), where the tabu list in the k-the iteration is denoted by TL_k.

$$x_{k+1} = argmin_{x \in N(x_k) \setminus TL_k} f(x) \tag{2}$$

This mechanism prevents the algorithm from revisiting previously explored regions of the search space, allowing TS to effectively escape local extrema and move toward more promising search space regions. Tabu list is updated in each iteration according to the First-In First-Out queue principle. The length of the tabu list, n, which is referred to as memory, is calibrated experimentally. Additionally, once every m_{int} iterations, a local search heuristic is applied to locate the nearest local extremum – this step is known as the intensification phase. Similarly, every m_{div} iterations, diversification is applied, where the algorithm jumps to a random region of the search space to assess whether it is, on average, better than the currently explored region. On top of that, we employ the strategy of aspiration moves, which is an exception that allows a normally forbidden (tabu) move to be accepted, if it results in a solution better than any previously found solution. Compared to SA, TS focuses more on exploitation, guiding the search faster toward promising search space regions, while SA emphasizes exploration, by probabilistically allowing moves that temporarily worsen solution quality. This exploitation-oriented strategy of TS often leads to faster convergence to high-quality solutions. Since TS employs a memory mechanism, through the use of a tabu list, it records recent moves and prohibits undoing them for a certain number of iterations. This prevents the algorithm from getting stuck in repetitive cycles around local extrema. In contrast, SA does not incorporate a memory mechanism, which may result in revisiting suboptimal regions of the search space multiple times.

4.3 Genetic algorithm

In contrast to SA and TS, GA processes in each iteration an n-element population of solutions $P_k = \{x_k^1, x_k^2, \dots, x_k^n\}$ rather than a single solution x_k. Additionally, unlike SA and TS, where a new solution is generated from the neighborhood of the current one $N(x_k)$, GA generates the next iteration's population of solutions P_{k+1} by employing consecutively three genetic operators, called selection, reproduction and mutation. Selection is responsible for choosing pairs of solutions from P_k for reproduction. We employed a tournament selection operator. During reproduction, for each pair of selected solutions (referred to as parents), GA randomly combines their elements, so that a new pair of solutions emerges (referred to as offspring). This process mimics natural selection and evolution. To ensure that each pair of parent solutions produces a new pair of feasible offspring solutions, we apply a set of crossover operations. For example, this involves taking routes from one solution x', removing cargo shipped along these routes from the second solution x'' and inserting these routes into the second solution x''.

After recombination, mutation is applied to introduce random changes to the offspring solutions. This helps maintain diversity in the population and prevents premature convergence to suboptimal search space regions. For mutation we employ operators analogous to perturbation operators used in SA and TS. After each offspring undergoes mutation with probability $p_m = 5\%$, GA produces the next iteration's population P_{k+1}. As a representation scheme to programmatically encode (represent) solutions as object instances, we use a natural representation, where all solution elements (routes, assigned vehicles etc.) are directly (explicitly) written into object instances storing them. In contrast, for example, to an interger programming formulation, where artificial variables are used to represent solutions.

5 Results of numerical simulations

In this section, we present the results of numerical simulations conducted to compare the efficiency of the three algorithms. The problem structure, including the logistic network and available resources, fully reflects the real-life scenario, however, the values of the operational parameters characterizing the problem, like cost related parameters and technical parameters, differ from those in the original case. To account for uncertainty arising from both the stochastic nature of the algorithms and the specific instance of the problem, the experiment was designed as follows:

1. The real-life problem instance was stochastically transformed into 10 variations, where parameter values related to operational costs and the technical characteristics of network nodes were perturbed.
2. Each of the 10 problem instances was solved 30 times by each algorithm, and the mean value trajectory of the objective function was computed across the 30 runs (for GA, this represents the mean trajectory of the best solution in the population).
3. Each algorithm was run on each problem instance for 8 hours (an operational time budget constraint).

Acta logistica - International Scientific Journal about Logistics
Volume: 12 2025 Issue: 2 Pages: 253-260 ISSN 1339-5629

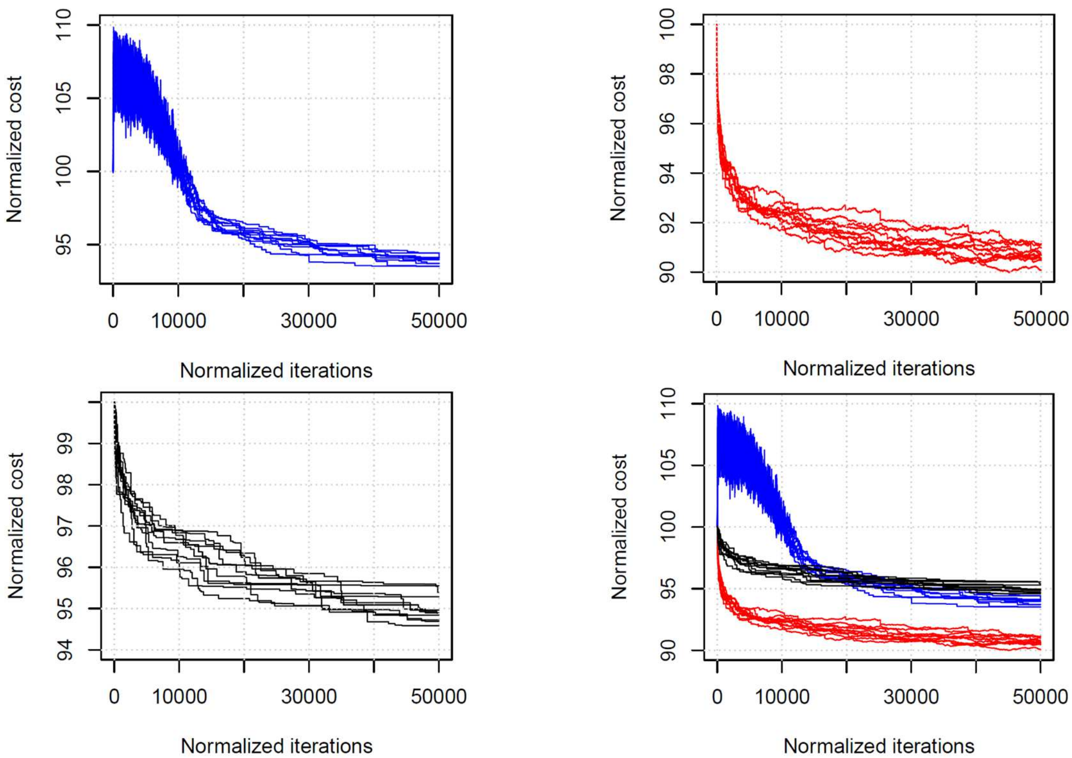

Figure 1 Mean trajectories of SA (upper left), TS (upper right) and GA (lower left). Mean trajectories of SA (blue), TS (red) and GA (black) imposed on a single plot (lower right)

Table 1 Statistics of costs at respective stages of search space exploration for each algorithm (Q1 = first quartile iteration, Q2 = median iteration, Q3 = third quartile iteration, Q4 = final iteration). Mean values are expressed in normalized terms, where the cost of the initial solution equals 100*. Standard deviations are expressed in percentage points. Means and standard deviations are calculated over 30 runs of 10 parametrizations of the considered postal optimization problem*

	Simulated Annealing	Tabu Search	Genetic Algorithm
Mean Q1	98.11	92.11	96.32
Std Q1	0.67	0.36	0.43
Mean Q2	95.30	91.34	95.61
Std Q2	0.42	0.58	0.31
Mean Q3	94.47	90.99	95.20
Std Q3	0.42	0.48	0.26
Mean Q4	93.91	90.68	94.98
Std Q4	0.29	0.32	0.32

The experiment was conducted on a server with 32 cores, enabling parallel computations (parallel runs of the algorithms). For better comparison, mean value trajectories of the objective functions are adjusted in the reported figures, so that the length of each trajectory equals 50,000 iterations (in reality, the computing time is the same across algorithms, while the number of iterations varies). Additionally, the objective function cost was normalized so that the cost of the initial solution equals 100 (for GA, this represents the cost of the best solution in the initial population). This means that if the solution cost decreases to 95, the algorithm has improved the solution by 5%. The initial solution was already pre-optimized, and in case of GA, the initial population consisted of the pre-optimized initial solution, solutions obtained by randomly perturbing the pre-optimized solution, and of purely random solutions.

Figure 1 presents the trajectories of SA, TS, and GA for the 10 variants of the postal optimization problem (see point 1 above). Due to the normalization of the objective function, these variants are directly comparable, both within each algorithm and across different algorithms. For better visualization, Figure 1 also overlays the trajectories onto a single plot (lower right plot).

Table 1 provides the average cost of solutions obtained by each algorithm at different stages of the optimization process, along with their standard deviations. Here, Q1 represents the first-quartile iteration, Q2 the median iteration, Q3 the third-quartile iteration, and Q4 the final iteration. The values in the table correspond to the trajectories shown in Figure 1. Since the cost of the initial pre-optimized solution is normalized to 100, the difference between 100 and the reported values represents the cost savings in percentage points.

The main conclusion drawn from the comparison is that the TS algorithm (presented in Figure 1 in red) produced the best results. It consistently outperformed the other two algorithms across all problem instances and throughout the entire search space exploration process. Regarding the final solutions, the SA algorithm (presented in Figure 1 in blue) ranked second, though the shape of its search trajectory differed significantly from that of the TS and GA (presented in Figure 1 in black). At the initial stages of the optimization process, SA consistently exhibited a tendency to sharply increase the solution cost, which aligns with its inherent mechanism – SA initially performs a highly exploratory search (a random search phase), frequently accepting moves that worsen the objective function value. As the optimization progresses, it becomes increasingly greedy, eventually converging to a fully greedy search state. The sharp cost increase observed in the early stages search space exploration by the SA algorithm can be mitigated by adjusting the initial value of the temperature parameter (t_0). However, experimental results indicate that for the considered postal optimization problem, achieving this would require forcing the algorithm into a highly greedy state early on, causing it to behave similarly to a local greedy descent method. Consequently, this leads to premature convergence and entrapment in a local minimum state, hindering global search space exploration. Despite this reported initial cost increase, SA, over time, demonstrates a significant potential for improving solutions. However, because it starts from an elevated cost level, it ultimately fails to catch up to TS, which does not exhibit such cost fluctuations in its early stages of the optimization process. The GA initially outperforms SA – for approximately 30% of the runtime, but ultimately it yields worse results than both SA and TS. This observation holds across different crossover operators tested in this study, as well as in scenarios where elitism mechanisms were incorporated.

Numerically, TS improved the initial pre-optimized solution by up to nearly 10% (with a standard deviation of 0.32 percentage points), compared to approximately 6% for SA (with standard deviation of 0.29 percentage points) and about 5% for the GA (with standard deviation of about 0.32 percentage points). In terms of cost dispersion in the final iteration, all three algorithms exhibited similar performance. Overall, in the considered case study, TS emerged as the most effective approach, with SA and GA ranking second and third, respectively. However, the difference between SA and GA was noticeably smaller than the difference between either of these two algorithms and TS. Using one-sided t-tests to compare the differences in mean values, both under the assumptions of equal and unequal variances, we find that the mean objective function value in the final iteration of the TS algorithm is statistically lower than that of the other two algorithms, with significance levels of 1%, 0.5%, and even 0.01%. Regarding the difference between the mean objective function values of SA and GA, the hypothesis that SA outperforms GA in the final iteration of the search space optimization process holds at the 1% significance level but must be rejected at the 0.5% significance level.

6 Conclusions

In this paper, we examined an optimization problem inspired by a real-life case study involving one of the largest postal service operators in Central and Eastern Europe. The objective was to determine whether popular metaheuristic optimization algorithms – Simulated Annealing (SA), Tabu Search (TS), and a Genetic Algorithm (GA) – could effectively reduce the cost of initially pre-optimized solutions within a reasonable timeframe of 8 hours, and to identify which algorithm performed best. The results indicate that all three algorithms achieved cost reduction, but to different extents. With fine-tuned parameters, SA initially increased the cost, diverging from the original solution before exhibiting a steep descent trajectory, ultimately producing solutions that were 5%–6% better than the originally pro-optimized one. As expected, due to its exploitative rather than explorative nature, TS produced a more uniformly decreasing trajectory, ultimately yielding solutions 9%–10% better than the initial one. GA ranked below SA and TS, following a uniformly decreasing trajectory similar to that of TS, and achieving cost reductions of 4%–5%.

One limitation of this study is the maximum runtime cap imposed on the algorithms, set at 8 hours. This runtime cap represents the operational requirement of the postal operator. Additional experiments suggest that the algorithms further improve the objective function, given more time, but our focus was on evaluating their performance within operational time constraints. Furthermore, the algorithm parameters were fine-tuned through a series of trial-and-error experiments. While this approach yielded effective configurations, employing external solvers to optimize parameter values could potentially enhance performance, which we leave for future investigations.

Acknowledgement
The state grant agency supported this article. The research, the results of which are presented in this paper, was supported by a National Centre for Research and Development grant no. POIR.04.01.04-00-0054/17-00.

References

[1] CHAU, M.L.Y., GKIOTSALITIS, K.: A systematic literature review on the use of metaheuristics for the optimisation of multimodal transportation, *Evolutionary Intelligence*, Vol. 18, No. 36, pp. 1-37, 2025. https://doi.org/10.1007/s12065-025-01020-2

[2] KOLOCH, G., LEWANDOWSKI, M., ZIENTARA, M., GRODECKI, G., MATUSZAK, P., KANTORSKI, I., NOWACKI, I.: A genetic algorithm for vehicle routing in logistic networks with practical constraints, *Statistical Review*, Vol. 68, No. 3, pp. 16-40, 2021. https://doi.org/10.5604/01.3001.0015.5584

[3] FERRELL, W., ELLIS, K., KAMINSKY, P., RAINWATER, C.: Horizontal collaboration: Opportunities for Improved Logistics Planning, *International Journal of Production Research*, Vol. 58, No. 14, pp. 4267-4284, 2019.

https://www.tandfonline.com/doi/full/10.1080/00207543.2019.1651457

[4] SEMBRING, N., TARIGAN, U., TIANA, Y.W., NASUTION, M.A.: Distribution improvement in achieving customer satisfaction, *Journal of Physics: Conference Series*, Vol. 1542, No. 1, pp. 1-8, 2020. https://iopscience.iop.org/article/10.1088/1742-6596/1542/1/012003

[5] VIVALDINI, M., PIRES, S.R.I., DE SOUZA, F.B.: Improving Logistic Services Through the Technology Used in Fleet Management, *Journal of Information Systems and Technology Management*, Vol. 9, No. 3, pp. 541-562, 2012. http://dx.doi.org/10.4301/s1807-17752012000300006

[6] MOR, A., SPERANZA, M.G.: Vehicle routing problems over time: a survey, *Annals of Operations Research*, Vol. 314, pp. 255-275, 2022. https://doi.org/10.1007/s10479-021-04488-0

[7] KONSTANTAKOPOULOS, G.D., GAYIALIS, S.P., KECHAGIAS, E.P.: Vehicle routing problem and related algorithms for logistics distribution: a literature review and classification, *Operational Research*, Vol. 22, pp. 2033-2062, 2022. https://doi.org/10.1007/s12351-020-00600-7

[8] KEDIA, R.K. NAICK, B.K.: *Review of vehicle route optimisation*, 2nd IEEE International Conference on Intelligent Transportation Engineering (ICITE) – Conference Paper, Singapore, pp. 57-61, 2017. https://doi.org/10.1109/ICITE.2017.8056881

[9] TOTH, P., VIGO, D.: *The Vehicle Routing Problem, Society for Industrial and Applied Mathematics*, Society for Industrial and Applied Mathematics, 2002. https://doi.org/10.1137/1.9780898718515

[10] GOLDEN, B., XINGYIN W., WASIL, E.: *The Evolution of the Vehicle Routing Problem*, A Survey of VRP Research and Practice from 2005 to 2022, Springer Cham, 2023. https://doi.org/10.1007/978-3-031-18716-2

[11] BRAEKERS, K., RAMAEKERS, K., VAN NIEUWENHUYSE, I.: The vehicle routing problem: State of the art classification and review, *Computers & Industrial Engineering*, Vol. 99, pp. 300-313, 2016. https://doi.org/10.1016/j.cie.2015.12.007

[12] CACERES-CRUZ, J., ARIAS, P., GUIMARANS, D., RIERA, D., JUAN, A.A.: Rich Vehicle Routing Problem: Survey, *ACM Computing Surveys*, Vol. 47, No. 2, pp. 1-28, 2014. https://doi.org/10.1145/2666003

[13] KIRKPATRICK, S., GELATT, C.D., VECCHI, M.P.: Optimization by Simulated Annealing, *Science*, Vol. 220, No. 4598, pp. 671-680, 1983. https://doi.org/10.1126/science.220.4598.671

[14] GLOVER, F.: Future Paths for Integer Programming and Links to Artificial Intelligence, *Computers and Operations Research*, Vol. 13, No. 5, pp. 533-549, 1986. https://doi.org/10.1016/0305-0548(86)90048-1

[15] HOLLAND, J.: *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*, The MIT Press, 1992. https://doi.org/10.7551/mitpress/1090.001.0001

[16] DRÉO, J., SIARRY, P., PÉTROWSKI, A., TAILLARD, E.: *Metaheuristics for Hard Optimization, Methods and Case Studies*, Heidelberg, Springer Berlin, 2006. https://doi.org/10.1007/3-540-30966-7

Review process

Single-blind peer review process.