https://doi.org/10.3311/PPtr.42468 | 1

Periodica Polytechnica Transportation Engineering

Testing the Efficiency of Flexible Activity Planning Solutions Using Activity Chains

Domokos Esztergár-Kiss[1*], Viktor Remeli[2]

[1] Department of Transport Technology and Economics, Faculty of Transportation Engineering and Vehicle Engineering, Budapest University of Technology and Economics, Műegyetem rkp. 3., 1111 Budapest, Hungary
[2] Department of Automotive Technologies, Faculty of Transportation Engineering and Vehicle Engineering, Budapest University of Technology and Economics, Műegyetem rkp. 3., 1111 Budapest, Hungary
[*] Corresponding author, e-mail: esztergar@mail.bme.hu

Received: 11 October 2025, Accepted: 12 February 2026, Published online: 28 August 2026

Abstract

Time-based Activity Chain Optimization (tACO) is a method for flexible, activity-based trip planning. It consists of finding a tour that visits exactly one location for each required activity while respecting time window constraints and returning to the starting location as early as possible. An activity can be performed at any one of several equally suitable locations, while not all locations are suitable for all activities. We compare our previous dynamic programming (DP) based exact tACO algorithm with our new mixed integer linear programming (MILP) model and show that using a straightforward formulation on a commercial solver we still cannot match the speed of the DP implementation. The experimental results obtained on 135 real-world problems underline the practical importance of our previous work and confirm the relevance of DP approaches on realistic problem sizes and in resource constrained environments, e.g. used in real-time trip planning systems and possibly running on embedded or mobile devices. The measurements also corroborate the assumed correctness of both implementations, as the optima coincide in each case (although the produced itineraries may vary considerably).

Keywords

activity chain optimization, mixed integer linear programming, dynamic programming

1 Introduction

Minimization of wasted resources is a central topic of traffic optimization that has gained both economic and environmental importance with globally increasing traffic loads. Meanwhile, a more localized demand for individual journey planning and optimization has also been ever present. From an individual traffic participant's perspective not only efficiency in planning, but various dimensions of flexibility and customization are critical as well. Depending on the specific use case, such flexibility dimensions might range from multi-modality, fleet logistics and dynamic weather and traffic information integration all the way to personalized, activity-centered planning.

Our work focuses on the planning of individual routes that enable time-efficient completion of a given set of activities. Prominent use-cases for our methods would involve personalized, activity-centered planning for the daily commute of individuals. A major application might be found in sightseeing tour planners for tourists who would like to make most efficient use of their precious time while staying in a foreign location and not knowing the exact places (only the types of places) they wish to visit, nor the best order or timing of the visits (that conforms both to personal preferences and opening hours, if applicable). It will be shown that the optimal solutions to the problem can vary substantially and often will necessarily include a fair amount of waiting time, creating thus an opportunity for unobtrusive partner promotion and related advertising schemes subtly delivered through the activity planner application's preferred (but still optimal) choices. In order to serve such and similar use-cases in a real-time and possibly embedded environment, the development of fast and memory-efficient algorithms is necessary. In this work we consider the viability of producing exact solutions to realistic activity planning problems.

The Time-based Activity Chain Optimization (tACO) problem, in its most basic form, is a Traveling Salesman Problem (TSP) variant characterized by both spatial flexibility and temporal constraints. Each required activity has to be performed at exactly one of the several equally

Cite this article as: Esztergár-Kiss, D., Remeli, V. (2026) "Testing the Efficiency of Flexible Activity Planning Solutions Using Activity Chains", Periodica Polytechnica Transportation Engineering, https://doi.org/10.3311/PPtr.42468

eligible alternative locations that are suitable for provision or hosting of named activity - similar to the Generalized TSP (GTSP, also called Set-TSP) problem. Meanwhile, both location specific time windows and user-given temporal preferences have to be observed simultaneously - similar to the TSP with Time Windows (TSP-TW) problem. The objective is to minimize the total time of the tour through the activities. The complete formal definition of the tACO problem will be given in Section 3.

Since the TSP is a special case of tACO, i.e. any TSP can be reduced to tACO by having exactly one location per activity and dilating the time windows such that they can be disregarded, we conclude that tACO belongs to the NP-hard family of problems (or NP-complete in the case of a decision problem formulation). With this work we deliver three main contributions:

- We give a new MILP formulation to the tACO problem that is suitable as an input for any general MILP solver.
- We evaluate the two distinct major approaches to tACO - dynamic programming (DP) and mixed integer linear programming (MILP) - by comparing the performance of our MILP model processed by a state-of-the-art commercial solver to our earlier DP algorithm on a set of 135 realistic, Budapest inner-city tACO problems.
- We briefly touch upon the issue of yet unexplored methods of delivering business value to the different stakeholders of activity planning systems through novel types of tACO-related advisory services and thus motivate further research into the topic. Most notably we identify the link with tourist trip design problems (Gavalas et al., 2014) and also with personalized, equi-optimal activity recommendation as a new vector for direct marketing.

2 Related work

2.1 Activity-based modeling

Activity-based modeling is a well-established modeling approach (Bastarianto et al., 2023) that represents how individuals plan and execute daily activities, rather than focusing on isolated trips. The method captures motivations, constraints, and the order of activities, allowing a more realistic simulation of travel behavior. As it reflects a full list of daily schedules of users, the modeling tool is especially valuable for evaluating mobility policies and emerging transportation services.

A recent paper presents a systematic literature review (Chau and Gkiotsalitis, 2025) on metaheuristic-based optimization techniques in the field of multimodal urban transportation, examining how such approaches address the computational complexity of large and high-dimensional networks. By classifying studies according to the problem scope, formulations, methods, and network settings, the results highlight trends and identify future research needs related to scalability, efficiency, and real-time deployment.

An activity-based travel demand model (Zhou et al., 2023) can be used to evaluate how new mobility services could reduce private car use. The results show that mobility hubs become effective when combined with vehicle sharing services, while improved public transport, better cycling infrastructure, and stricter parking policies further decrease car trips and support more sustainable travel patterns.

Another study (Adenaw and Bachmeier, 2022) introduces a method for generating activity-based mobility plans by combining existing trip-based models with mobility survey data, avoiding the complexity of activity-based demand models. The approach produces realistic mobility patterns at both microscopic and macroscopic levels, making it a practical alternative when full activity-based modeling is not feasible.

Finally, another study (Mahdi and Esztergár-Kiss, 2023) develops an activity-based modeling approach combined with a genetic algorithm to generate personalized tourist itineraries that account for individual preferences, realistic constraints, and flexible activity time windows. The results of private vehicle and public transport usage scenarios show significant travel time reductions and performance benefits, demonstrating that the method effectively supports tourists in solving scheduling and mobility planning problems.

2.2 Traveling Salesman Problem (TSP)

The Traveling Salesman Problem's mathematical properties have been studied extensively in the last 65 years, starting with the 1954 seminal work of Dantzig et al. (1954), who developed a mixed integer linear programming formulation for the TSP and, using some instance specific heuristics, arrived at a provably optimal solution for the 49 city tour across the U.S. The MILP formulation and the related simplex-based (Dantzig et al., 1955) algorithms that involve methodical generation of on-demand relaxation cuts as well as choosing efficient branch and bound strategies have gone through a lot of sophistication throughout the decades (Applegate et al., 2003), yet remain essentially in the same line of thought as the original approach while also being the absolute record-holders in the TSP domain, with the Concorde solver tackling an 85,900 nodes problem in April 2006 (Applegate et al., 2009) and a 109,399 nodes problem in September 2017.

Since Dantzig-Fulkerson-Johnson's MILP formulation (the so-called DFJ model) assumes an exponential number of subtour elimination constraints (with regards to the number of nodes), it can only be feasibly managed by on-demand integration of these constraints in a sophisticated branch-and-cut setup. A more accessible model is the Miller-TuckerZemlin (MTZ) formulation (Miller et al., 1960), which manages to pack subtour eliminations into a polynomial number of constraints and a linear number of auxiliary variables. Even though its linear relaxation is somewhat weaker than that of DFJ (the MTZ polytope in fact subsumes the DFJ polytope (Velednitsky, 2017)), due to the significantly smaller number of constraints and resulting memory and runtime efficiency the MTZ is a popular choice for use with generic MILP solvers. The inherent ordering of visited nodes of MTZ lends itself well to expressing time window constraints and was therefore chosen as a basis for our formulation given in Section 3. There exist several other types of models known for the TSP, most notably the Multi Commodity Flow formulation (Öncan et al., 2009), and specialized formulations for the GTSP (Pop, 2007) and the TSP-TW (Kara and Derya, 2015), which served as a starting point for our own model.

The popularity of the MILP approach was broken only for a few years in the early 1970s (Cook, 2012), when the dynamic programming algorithm discovered independently by Bellman (1962) and Held and Karp (1962) seemed to overtake linear programming in practical performance while also delivering a much better worst-case runtime guarantee of $O(2^n n^2)$ steps, n being the number of cities. The dynamic programming tACO algorithm we developed in our previous work (Esztergár-Kiss and Remeli, 2021) is essentially an adaptation of the Bellman-Held-Karp principles and builds upon a similarly formulated optimization invariant while guaranteeing a worst-case runtime complexity of $O(2^m n^2)$, m being the number of activities and n being the number of locations.

A short overview of our Held-Karp inspired ExtACO (Esztergár-Kiss and Remeli, 2021) dynamic programming algorithm for Time-based Activity Chain Optimization is given here.

Problem size parameters:

- m: number of activity clusters including home
- n: number of location nodes including home
- Runtime complexity: $O(2^m n^2)$.
- Memory complexity: $O(2^m n)$.
- Main advantage: in practice, we usually have $m \ll n$.
- Optimization invariant: If there is a solution, then there exists at least one time-optimal solution whose subsolutions are all time-optimal.
- We can forward-construct such a special solution using the Held-Karp method (Held and Karp, 1962).

2.3 Activity Chain Optimization (ACO)

Intelligent activity planning methods, notably the activity chains-based modeling of daily travel patterns have seen a significant increase in research interest in the last 16 years (Miller and Roorda, 2003; Timmermans et al., 2003). A foundational book summarizing the state of the art was published in 2005 by Timmermans (2005). The TSP is the best studied problem model for activity planning and forms the basis for our discussion. A review of metaheuristic algorithms for solving TSP-based scheduling optimization problems was presented, where it turns out that GA is the most applied algorithm in publications, but ACO is the most cited one (Toaza and Esztergár-Kiss, 2023).

The ACO problem family, to which our particular problem variant (tACO) belongs, was first described by Esztergár-Kiss et al. (2017) in 2017, who give a general framework for first solving the ACO problem as a TSP-TW instance with a predefined choice of locations for the activities ("basic scenario"), which is then later relaxed towards the original problem by reintroducing spatial flexibility. They also implement this framework using a two-stage heuristic approach based on genetic algorithms, which are shown to work on small multi-modal examples on a real map of Budapest (Esztergár-Kiss et al., 2018).

A MILP model similar to ours - but not suited for the tACO problem variant - is given in the recent work of Yuan et al. (2020). While both formulations employ auxiliary node selection variables as described in Pop's GTSP modeling paper (Pop, 2007), our model differs in three notable ways:

1. our model accounts for the possibility of activity durations and wait times (arrival before commencement of activity);
2. we use a different objective function formulation in order to be able to optimize the dynamic cost (incurred with waiting times which depend on the preceding part of the tour, as described in Section 3);
3. we use a different MTZ ordering for subtour elimination that supports the dynamic total time objective.

The proposed MILP model offers several advantages compared to alternative approaches, even though it is less computationally efficient than DP for large instances. The MILP formulation (Spyrou et al., 2025) can easily incorporate additional constraints and new objective functions, making it adaptable to a wide range of real-world scenarios. The model can be solved using widely available commercial and open-source optimization solvers, facilitating reproducibility and practical implementation. When solvable within reasonable time, the MILP model guarantees globally optimal solutions, which is valuable for benchmarking of small-to-medium-sized problems. Finally, the solution provides a mathematically transparent and well-structured framework, which is useful for theoretical analysis, validation, and comparison with other optimization methods.

3 Developed model

3.1 Formal definition of the tACO problem

The tACO problem can formally be defined as follows:

Find a minimum $\tilde{t}_{0'}$ finite finishing time Hamiltonian cycle across the disjunct non-empty activity partitions $A_0 = \{0\}, A_1, ..., A_{m-1}$ of location vertices $V = \{0, 1, ... ,n-1\}$ of a directed graph $G = (V, E)$ starting from a designated home activity $A_0 = \{0\}$ at a given time $\tilde{t}_0$ and traversing the graph in such a way that exactly one vertex of each activity cluster is visited and no time window constraints are violated. Each location vertex $i \in V$ is associated with a time window interval $TW_i = \left[w_i', w_i''\right], 0 \le w_i' \le w_i''$ and an activity duration $\delta_i \ge 0$. Only inter-partition edges $E = \left\{\left(i,j\right) : i \in A_u, j \in Av, u \ne v\right\}$ need be considered, each of which can be mapped to a traveling time $T_{ij} \in R_{0+}$ between vertices i and j. The total time cost $C_{ij} = \tilde{t}_j - \tilde{t}_i$ incurred by performing the activity at i and then traversing an edge (i, j) is dynamically (recursively) defined by the preceding part of the tour ($\hat{t}$ denotes arrival times while $\tilde{t}$ denotes activity starting times with $\tilde{t}_0$ marking the time of departure and $\tilde{t}_{0'}$ marking the time of finishing the tour at the home node):

$\tilde{t}_0 \ge 0$ is given and $\delta_0 = \delta_{0'} = 0$

$$\hat{t}_j = \tilde{t}_i + \delta_i + T_{ij}$$

$$\tilde{t}_j = \begin{cases} w_j' \text{ if } \hat{t}_j < w_j' \le w_j'' - \delta_j \\ \hat{t}_j \text{ if } w_j' \le \hat{t}_j \le w_j'' - \delta_j \\ \infty \text{ otherwise} \end{cases} .$$

The above relation ensures that time window constraints are respected in all possible minimum $\tilde{t}_{0'}$ finite finishing time tours which (if they exist) are also equivalent to the minimum total cost tours where total cost equals $C = \sum_{(i,j)\in Tour} C_{ij}$.

Note that travel between two locations in a given direction is assumed to always take a fixed amount of time (T_{ij}), and performing an activity has a specific duration as well that in the basic formulation of the problem we shall assume to be location-independent, thus:

$$\delta_i = \delta_j \forall i, j : A_{[i]} = A_{[j]} .$$

If an activity is performed at a specific location, then it must both commence and end within a single, continuous time window that is obtained by the intersection of user defined and location-specific intervals. Also note that it is possible to arrive to a location earlier than the time window and then wait before commencing the activity at the beginning of the time window. If a real-world location offers several activities, it can be modeled as several locations at the same spot with (near) zero travel times within the group. If the real-world problem specification requires the completion of the same activity more than once (maybe in different locations or time-intervals), then this should be modeled as different activities.

3.2 MILP model for tACO

Using the notation introduced above, we define our MILP model as follows:

Minimize $\tilde{t}_{0'}$

s.t.

$$\sum\nolimits_{i\in A_u} y_i = 1 \quad \forall u \in \{0,\dots,m-1\} , \tag{1}$$

$$\sum\nolimits_{j\in V\setminus\{i\}} x_{ij} = \sum\nolimits_{j\in V\setminus\{i\}} x_{ji} = y_i \quad \forall i \in V , \tag{2}$$

$$w_i' y_i \le \tilde{t}_i \le \left(w_i'' - \delta_i\right) y_i \quad \forall i \in V , \tag{3}$$

$$\tilde{t}_j - \tilde{t}_i \ge \delta_i y_i - w_0'' + \left(w_0'' + T_{ij}\right) x_{ij} \quad \forall i, j \in V, i \ne j, i \ne 0 , \tag{4}$$

$$\tilde{t}_i \ge \left(\tilde{t}_0 + T_{0i}\right) x_{0i} \quad \forall i \in V, i \ne 0 , \tag{5}$$

$$y_i \in \{0,1\} \quad \forall i \in V , \tag{6}$$

$$x_{ij} \in \{0,1\} \quad \forall i, j \in V, i \ne j . \tag{7}$$

The model has $n^2 + 3n + m$ constraints (besides integrality) defined on $n^2 + n$ decision variables: $y_i \in \{0,1\}, x_{ij} \in \{0,1\}$, and $\tilde{t}_i \in R_{0+}, i, j \in V, i \ne j$. Note that $\tilde{t}_{0'}$ is a variable and $\tilde{t}_0$ a parameter. The first constraint group ensures that only one location is selected for each activity. The second constraint group consists of two equalities that together guarantee that the selected locations will have exactly one incoming and one outgoing arc selected for the tour, while the remaining locations will have no arcs selected. The third constraint group ensures that time windows are respected. The fourth constraint group gives an MTZ-like temporal ordering of selected locations that takes into account travel times and activity durations while also serving a subtour elimination purpose. The fifth constraint group supplements the fourth one by setting a starting point for an absolute (instead of a relative) temporal ordering and thus supporting the minimization of the tour's completion time with a lower bound. The sixth and seventh constraint groups ensure the solution is binary where necessary.

4 Experimental method

We did a comparative evaluation of our DP algorithm (ExtACO) and our MILP model given in the Section 3.2 (MilptACO) on 135 realistic activity chain problems of different sizes.

Budapest road network data and facilities' types and opening hours data was obtained from the OpenStreet Map database (Haklay and Weber, 2008). The travel times matrices were derived using a locally installed OpenTripPlanner (Hillsman and Barbeau, 2011) server instance. For technical reasons the chosen modality for calculating interlocation travel times in minutes was walking. Many locations were unreachable with other modalities, and this prevented us from building a full travel times graph without excessive manual intervention. Note that the choice of modality does not affect the conclusions regarding the performance of the algorithms.

We created three different maps of Budapest, all of which offer 20 different activities in 100, 300 and 1,000 possible locations (corresponding to 5, 15 and 50 locations per activity). The three maps will be abbreviated as 05pa, 15pa and 50pa map, respectively.

A problem specification (like the one given in Table 1) together with the pre-calculated map data fully defines a tACO problem instance. The first row of the specification gives the location of the starting point (home activity), the time of the start of the tour and the latest finishing time. The remaining rows list the required activities in arbitrary order, with a planned duration (in minutes) and a preferred time window specified for each of them. Entirely new kinds of activities (not defined on the map) can also be added as so called "fixed demand points" – i.e. activities with a single location defined in the specification.

The user given time window and the location-specific opening hours must have a non-empty intersection which is longer than the activity duration for a location to be considered feasible. All infeasible locations are removed from the problem by default before being processed by a solver. If any activity remains without locations, the whole problem is infeasible.

We manually prepared 45 realistic ACO specifications which were combined with the three maps for a total of 135 problem instances. The problems represent daily commute planning tasks of practical sizes (5-9 activities besides home) with one spatially fixed activity ("Work"), and all other non-home activities spatially flexible.

Both algorithms were evaluated on all 135 problems. The test runs were performed on an Intel® Core™ i7-8700 CPU @ 3.20GHz×12 64-bit system with 16 GB of RAM. The MILP model of the problem instances was fed to 2 open source (GLPK-MI and CoinOr CBC) and 2 commercial (Gurobi and Xpress) solvers. The results obtained with Xpress are reported, as it seems to be the most suitable solver for this particular model.

We measured the execution time of computations and the generated tour costs (we define the tour cost as wasted time: the sum total of travel times and wait times). Since activity durations are location-independent, the sum of durations is constant for a given problem, therefore the minimum wasted time solution is also a minimum finishing time solution (and *vice versa*).

5 Results

First, we give an example of a tACO instance solved by both ExtACO and MilptACO solvers, where ExtACO provided a solution within 0.02 s, while MilptACO did it in 23.89. The specification is the same as in Table 1, and the 15pa map was used for constructing the instance. The differences between the two optimal solutions can be observed in Fig. 1.

Table 1 An example tACO problem instance specification with activities

Activity	Dur.	DemandStart	DemandClose	FixedLat.	FixedLon.
Start/Goal	0	2025.04.15 6:30	2025.04.15 23:59	47.51 4895	19.03 2784
Bakery	5	2025.04.15 6:30	2025.04.15 10:00		
Work	480	205.04.15 6:30	2025.04.15 20:00	47.50 7163	19.04 7851
Grocery Store	15	2025.04.15 7:00	2025.04.15 22:00		
Cosmetics Shop	20	2025.04.15 8:00	2025.04.15 18:00		
Pub	90	2025.04.15 18:00	2025.04.15 23:59		

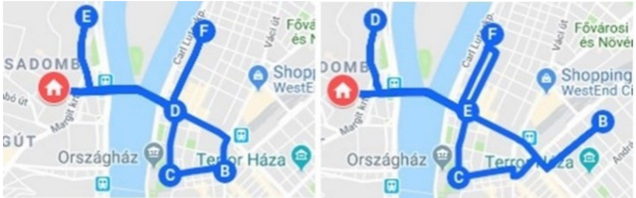

Fig. 1 Two optimal solutions to a 5-activity tACO instance on a map

It can be clearly seen that due to temporal preferences specified by the user, significant amounts of wasted time are inevitably incurred, and this wasted time carries over to all optimal solutions. Consider the "Pub" activity requirement, for instance. Since the user is not willing to start the "Pub" activity before 18:00, they have to spend some time uselessly, and also gain the opportunity to wander off to farther locations without affecting optimality. It is evident that some sort of activity advisory system might be of benefit to both users and advertisers in these (probably rather frequent) cases of inherently wasteful daily schedule specifications.

Within a 10 minutes per problem timeframe, MilptACO was able to solve only 54 optimization tasks out of 135 (40%), having solved none of the 50pa map problems and not having solved all of the 05pa map problems. Meanwhile ExtACO solved all 135, each within 4 seconds. The numbers of solved instances are given in greater detail in Table 2. Cases marked with bold terminated with out of memory error even before reaching the 10-minute timeout (m-1 stands for the number of non-home activities).

Table 2 Number of tACO problem instances solved before timeout (given as a ratio)

Map	m-1	ExtACO	MilptACO
05pa	5	9/9	9/9
	6	9/9	9/9
	7	9/9	9/9
	8	9/9	9/9
	9	9/9	0/9
15pa	5	9/9	9/9
	6	9/9	9/9
	7	9/9	0/9
	8	9/9	0/9
	9	9/9	0/9
50pa	5	9/9	0/9
	6	9/9	0/9
	7	9/9	0/9
	8	9/9	0/9
	9	9/9	0/9
Total		135/135	54/135

The MILP solver struggles with large tACO cases because the time window induced waiting creates many equivalent optimal or near-optimal solutions, greatly expanding the search space. As the number of activities and feasible locations grows, the MILP formulation must explicitly represent and explore these combinations, leading to excessive branching and high memory consumption. In contrast, the DP-based solution implicitly prunes dominated and redundant cases early, allowing it to remain fast and memory-efficient even on large and realistic instances.

The experimental results clearly show that the MilptACO formulation has neither speed nor memory advantage over our dynamic programming ExtACO algorithm, at least not in the examined problem size ranges.

In Fig. 2, the computation times of 54 tACO problem instances solved by MilptACO and all 135 instances solved by ExtACO are presented. The execution time increases with an increasing number of locations and/or activities. Note that many of the 135 ExtACO measurements coincide and cannot be distinguished in the scatterplot. The lines connecting the averages of the 05pa, 15pa and 50pa maps' measurements are for visual aid only.

6 Discussion and conclusion

Since the solutions obtained by both our approaches are of the same quality (in both cases optimal, but not necessarily the exact same itinerary), the only important difference arises in the runtime performance and applicability of the algorithms. Our interest lies in devising efficient solving methods for realistic tACO problem instances, which in the case of daily commutes typically range between 5–9 activities and dozens to hundreds of locations. Combinatorial optimization problems of this kind are usually first approached with a naive MILP formulation and given to a generic solver, which can often give surprisingly good results (further details can be found in the Appendix).

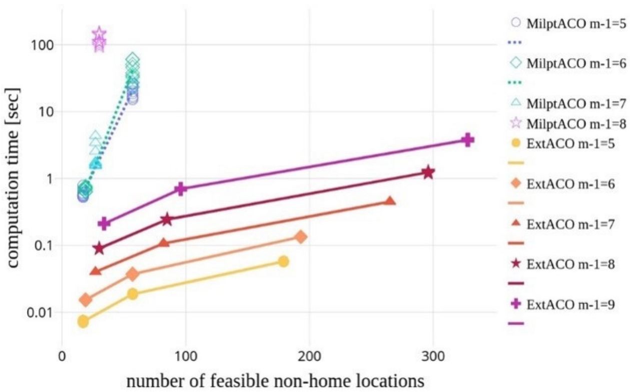

Fig. 2 Comparison of the results

We have shown that for the tACO problem, this is not the case. In this paper we gave a new MILP formulation for the tACO problem, and it was shown not to be competitive with our previous algorithm even with a MILP solver. There remains a theoretical possibility that with sufficient memory resources we could scale to a problem size where the MILP solver would outperform the DP algorithm even using the same formulation. This, however, is not the scope of our study, as from the viewpoint of solving real-world problem instances in a real-time and possibly embedded environment, any possible gain in the very large instance / very large memory domain remains irrelevant.

We hypothesize however, that further improvements to the MILP model and introduction of problem-specific branch-and-cut strategies might lead to substantial speed benefits on somewhat larger problem instances (12–20 activities) which are currently out of reach for the DP implementation, but may still find practical use somewhere, e.g. activity planning several days ahead, flexible last-mile delivery services, or even trajectory planning for robotics.

Moreover, the MILP model can be significantly improved by integrating advanced optimization techniques and elaborating hybrid solutions with heuristic methods, especially to address scalability and runtime limitations. Potential improvement directions include decomposition techniques, hybrid approaches, preprocessing and variable pruning, solver-level enhancements, and integration with machine learning-based methods. By combining MILP with heuristics, decomposition, and learning-based strategies, the model can retain its flexibility and optimality guarantees, while achieving much better scalability and computational efficiency. Such hybrid frameworks would be particularly well suited for real-time multimodal trip planning and large-scale urban optimization problems.

Although we focused on activity planning use-cases for passengers, we can assume that the greatest economic impact of the ACO approach proposed in this paper will be found in last-mile delivery logistics as soon as the method becomes viable for larger problem sizes. A typical delivery roundtrip consists of several dozen packages with a few possible locations and time windows defined by each customer that is clearly a problem best modeled as ACO. Today the customer either must wait at home or make a detour to a pick-up point without further realistic alternatives. In the future by applying the ACO framework, the customer can realize the pre-planned daily routine, while the delivery will find a suitable location within the activity chain to deliver the packages. Since the problem is global and the solution is scalable in nature, even a 5% efficiency increase would amount to a huge business opportunity considering the size of the rapidly growing last-mile delivery sector that was 197 billion USD in 2025.

When talking about increasing freight efficiency, we want to be clear that our approach allows for easy adaptation to different optimization criteria. Thus, we may optimize not only for time, but also for fuel consumption, delivery reliability, and resource utilization. Therefore, our method can be aligned with broader policy goals, such as reducing urban congestion and emissions. The technological framework for dynamically synchronizing customer needs and constraints with freight companies is an IT challenge that can be solved either by individual stakeholders (delivery service and pickup point companies) or by a third-party provider delivering standardized interfaces, meeting data protection and other necessary regulations, while possibly acting as a broker (intermediary). This would allow smaller logistics operators to also take part in the world of advanced delivery services.

Acknowledgement

The research was supported by OTKA FK23 145899 project funded by the National Research, Development and Innovation Office.

References

Adenaw, L., Bachmeier, Q. (2022) "Generating Activity-Based Mobility Plans from Trip-Based Models and Mobility Surveys", Applied Sciences, 12(17), 8456.
https://doi.org/10.3390/app12178456

Applegate, D., Bixby, R., Chvátal, V., Cook, W. (2003) "Implementing the Dantzig-Fulkerson-Johnson algorithm for large traveling salesman problems", Mathematical Programming, 97(1), pp. 91–153.
https://doi.org/10.1007/s10107-003-0440-4

Applegate, D. L., Bixby, R. E., Chvátal, V., Cook, W., Espinoza, D. G., Goycoolea, M., Helsgaun, K. (2009) "Certification of an optimal TSP tour through 85,900 cities", Operations Research Letters, 37(1), pp. 11–15.
https://doi.org/10.1016/j.orl.2008.09.006

Bastarianto, F. F., Hancock, T. O., Choudhury, C. F., Manley, E. (2023) "Agent-based models in urban transportation: review, challenges, and opportunities", European Transport Research Review, 15(1), 19.
https://doi.org/10.1186/s12544-023-00590-5

Bellman, R. (1962) "Dynamic Programming Treatment of the Travelling Salesman Problem", Journal of the ACM, 9(1), pp. 61–63. https://doi.org/10.1145/321105.321111

Chau, M. L. Y., Gkiotsalitis, K. (2025) "A systematic literature review on the use of metaheuristics for the optimisation of multimodal transportation", Evolutionary Intelligence, 18(2), 36. https://doi.org/10.1007/s12065-025-01020-2

Cook, W. J. (2012) "In Pursuit of the Traveling Salesman: Mathematics at the Limits of Computation", Princeton University Press. ISBN: 9780691163529

Dantzig, G., Fulkerson, R., Johnson, S. (1954) "Solution of a Large-Scale Traveling-Salesman Problem", Journal of the Operations Research Society of America, 2(4), pp. 393–410. https://doi.org/10.1287/opre.2.4.393

Dantzig, G. B., Orden, A., Wolfe, P. (1955) "The generalized simplex method for minimizing a linear form under linear inequality restraints", Pacific Journal of Mathematics, 5(2), pp. 183–195. https://doi.org/10.2140/pjm.1955.5.183

Esztergár-Kiss, D., Remeli, V. (2021) "Towards practical algorithms for activity chain optimization", Transportation Letters, 13(1), pp. 64–76. https://doi.org/10.1080/19427867.2019.1702250

Esztergár-Kiss, D., Rózsa, Z., Tettamanti, T. (2017) "Comparative analysis of test cases of the activity chain optimization method", Transportation Research Procedia, 27, pp. 286–293. https://doi.org/10.1016/j.trpro.2017.12.136

Esztergár-Kiss, D., Rózsa, Z., Tettamanti, T. (2018) "Extensions of the Activity Chain Optimization Method", Journal of Urban Technology, 25(2), pp. 125–142. https://doi.org/10.1080/10630732.2017.1407998

Gavalas, D., Konstantopoulos, C., Mastakas, K., Pantziou, G. (2014) "A survey on algorithmic approaches for solving tourist trip design problems", Journal of Heuristics, 20(3), pp. 291–328. https://doi.org/10.1007/s10732-014-9242-5

Haklay, M., Weber, P. (2008) "OpenStreet Map: User-Generated Street Maps", IEEE Pervasive Computing, 7(4), pp. 12–18. https://doi.org/10.1109/MPRV.2008.80

Held, M., Karp, R. M. (1962) "A Dynamic Programming Approach to Sequencing Problems", Journal of the Society for Industrial and Applied Mathematics, 10(1), pp. 196–210. https://doi.org/10.1137/0110015

Hillsman, E. L., Barbeau, S. J. (2011) "Enabling Cost-Effective Multimodal Trip Planners through Open Transit Data", National Center for Transit Research, Florida, United States, Report No. CUTR-NCTR-RR-2010-05. https://doi.org/10.5038/CUTR-NCTR-RR-2010-05

Kara, I., Derya, T. (2015) "Formulations for Minimizing Tour Duration of the Traveling Salesman Problem with Time Windows", Procedia Economics and Finance, 26, pp. 1026–1034. https://doi.org/10.1016/S2212-5671(15)00926-0

Mahdi, A. J., Esztergár-Kiss, D. (2023) "Supporting scheduling decisions by using genetic algorithm based on tourists' preferences", Applied Soft Computing, 148, 110857. https://doi.org/10.1016/j.asoc.2023.110857

Miller, C. E., Tucker, A. W., Zemlin, R. A. (1960) "Integer Programming Formulation of Traveling Salesman Problems", Journal of the ACM, 7(4), pp. 326–329. https://doi.org/10.1145/321043.321046

Miller, E. J., Roorda, M. J. (2003) "Prototype Model of Household Activity-Travel Scheduling", Transportation Research Record: Journal of the Transportation Research Board, 1831(1), pp. 114–121. https://doi.org/10.3141/1831-13

Öncan, T., Altınel, I. K., Laporte, G. (2009) "A comparative analysis of several asymmetric traveling salesman problem formulations", Computers & Operations Research, 36(3), pp. 637–654. https://doi.org/10.1016/j.cor.2007.11.008

Pop, P. C. (2007) "New Integer Programming Formulations of the Generalized Travelling Salesman Problem", American Journal of Applied Sciences, 4(11), pp. 932–937. https://doi.org/10.3844/ajassp.2007.932.937

Spyrou, E. D., Kappatos, V., Gkemou, M., Bekiaris, E. (2025) "Multimodal Transport Optimization from Doorstep to Airport Using Mixed-Integer Linear Programming and Dynamic Programming", Sustainability, 17(17), 7937. https://doi.org/10.3390/su17177937

Timmermans, H. J. P. (2005) "Progress in activity-based analysis", Elsevier. ISBN: 9780080445816

Timmermans, H., van der Waerden, P., Alves, M., Polak, J., Ellis, S., Harvey, A. S., Kurose, S., Zandee, R. (2003) "Spatial context and the complexity of daily travel patterns: an international comparison", Journal of Transport Geography, 11(1), pp. 37–46. https://doi.org/10.1016/S0966-6923(02)00050-9

Toaza, B., Esztergár-Kiss, D. (2023) "A review of metaheuristic algorithms for solving TSP-based scheduling optimization problems", Applied Soft Computing, 148, 110908. https://doi.org/10.1016/j.asoc.2023.110908

Velednitsky, M. (2017) "Short combinatorial proof that the DFJ polytope is contained in the MTZ polytope for the Asymmetric Traveling Salesman Problem", Operations Research Letters, 45(4), pp. 323–324. https://doi.org/10.1016/j.orl.2017.04.010

Yuan, Y., Cattaruzza, D., Ogier, M., Semet, F. (2020) "A branch-and-cut algorithm for the generalized traveling salesman problem with time windows", European Journal of Operational Research, 286(3), pp. 849–866. https://doi.org/10.1016/j.ejor.2020.04.024

Zhou, H., Dorsman, J. L., Mandjes, M., Snelder, M. (2023) "Sustainable mobility strategies and their impact: a case study using a multimodal activity based model", Case Studies on Transport Policy, 11, 100945. https://doi.org/10.1016/j.cstp.2022.100945

Appendix

>>> ExtACO solving tACO Problem: 5 activities, 57 nodes (+ home).

Itinerary SOLVED for Problem: 5 activities, 57 nodes (+ home). Comp. time: 0.019 sec. Wasted time: 196 mins.

00. Start/Goal [0|None] Home (47.514895, 19.032784) [06:30-23:59] left at 06:30

01. Bakery [16|49] Norbi Update P´eks´eg (47.50750479999999, 19.0551639) [07-18]

Arrived|Entered|Left-At|Tr.Time|TrTotal|Tr+wait|Travel+Wait Total

06:59 07:00 07:05 00:29 00:29 00:30 00:30
419 420 425 29 29 30 30

02. Work [57|None] location_62 (47.507163, 19.047851) [06:30-20:00]

Arrived|Entered|Left-At|Tr.Time|TrTotal|Tr+wait|Travel+Wait Total

07:13 07:13 15:13 00:08 00:37 00:08 00:38
433 433 913 8 37 8 38

03. Cosmetics Shop [48|157] Lush (47.512708, 19.0485549) [10:00-19:00]

Arrived|Entered|Left-At|Tr.Time|TrTotal|Tr+wait|Travel+Wait Total

15:22 15:22 15:42 00:09 00:46 00:09 00:47
922 922 942 9 46 9 47

04. Grocery Store [28|122] CBA (Bet-Land) (47.5211423, 19.0367746) [06-22]

Arrived|Entered|Left-At|Tr.Time|TrTotal|Tr+wait|Travel+Wait Total

16:04 16:04 16:19 00:22 01:08 00:22 01:09
964 964 979 22 68 22 69

05. Pub [1|31] Pozsonyi S¨or¨oz}o (47.519873, 19.0524686) [14:00-22:00]

Arrived|Entered|Left-At|Tr.Time|TrTotal|Tr+wait|Travel+Wait Total

16:50 18:00 19:30 00:31 01:39 01:41 02:50
1010 1080 1170 31 99 101 170

06. Start/Goal [0|None] Home (47.514895, 19.032784) [06:30-23:59]

Arrived|Entered|Left-At|Tr.Time|TrTotal|Tr+wait|Travel+Wait Total

19:56 19:56 19:56 00:26 02:05 00:26 03:16
1196 1196 1196 26 125 26 196

>>> MilpACO solving tACO Problem: 5 activities, 57 nodes (+ home).

Itinerary SOLVED for Problem: 5 activities, 57 nodes (+ home). Comp. time: 23.891 sec. Wasted time: 196 mins.

00. Start/Goal [0|None] Home (47.514895, 19.032784) [06:30-23:59] left at 06:30

01. Bakery [15|48] nan (47.51197, 19.06706) [06:00-22:00]

Arrived|Entered|Left-At|Tr.Time|TrTotal|Tr+wait|Travel+Wait Total

07:09 07:09 07:14 00:39 00:39 00:39 00:39
429 429 434 39 39 39 39

02. Work [57|None] location_62 (47.507163, 19.047851) [06:30-20:00]

Arrived|Entered|Left-At|Tr.Time|TrTotal|Tr+wait|Travel+Wait Total

07:38 07:38 15:38 00:24 01:03 00:24 01:03
458 458 938 24 63 24 63

03. Grocery Store [28|122] CBA (Bet-Land) (47.5211423, 19.0367746) [06-22]

Arrived|Entered|Left-At|Tr.Time|TrTotal|Tr+wait|Travel+Wait Total

16:09 16:09 16:24 00:31 01:34 00:31 01:34
969 969 984 31 94 31 94

04. Cosmetics Shop [53|162] Rossmann (47.51300000000001, 19.048974) [08-20]

Arrived|Entered|Left-At|Tr.Time|TrTotal|Tr+wait|Travel+Wait Total

16:46 16:46 17:06 00:22 01:56 00:22 01:56
1006 1006 1026 22 116 22 116

05. Pub [1|31] Pozsonyi S¨or¨oz}o (47.519873, 19.0524686) [14:00-22:00]

Arrived|Entered|Left-At|Tr.Time|TrTotal|Tr+wait|Travel+Wait Total

17:18 18:00 19:30 00:12 02:08 00:54 02:50
1038 1080 1170 12 128 54 170

06. Start/Goal [0|None] Home (47.514895, 19.032784) [06:30-23:59]

Arrived|Entered|Left-At|Tr.Time|TrTotal|Tr+wait|Travel+Wait Total

19:56 19:56 19:56 00:26 02:34 00:26 03:16
1196 1196 1196 26 154 26 196