Emission-Reducing Vehicle Routing in Food Logistics

Armin Wolf [1] and Silke Cuno [2]

Abstract: A web service for emission-reducing vehicle routing in food logistics is presented. This service computes emission-reduced round trips for the transportation of food products from (and to) a depot to (and from) regional locations like food stores or food hubs. Further, this service considers the travelling times between locations and the produced greenhouse gas emissions of the according drives as well as the time windows needed for pickup and delivery. The loading capacities of the vehicles and an efficient "last-in-first-out" packaging order of the vehicles are respected, with the consequence that goods picked-up last should be delivered and unloaded first. This is to avoid unnecessary work.

Keywords: Artificial Intelligence; Vehicle Routing; Food Logistics; Sustainability; Operations Research; Graph Theory; Resource Scheduling Problem; Constraint Optimisation; Travelling Salesperson Problem

1 Introduction

Within the research and development project "Stadt-Land-Fluss" (SLF) in the field of regional food systems, funded by the German Federal Ministry of Food and Agriculture, an AI application for transport optimisation in food logistics was developed: a web service that computes emission-reduced round trips for pickup and delivery tasks of goods, mainly food.

The objective of the SLF emission-reducing vehicle routing service is to minimise the total emissions of greenhouse gases caused by the daily journeys of a regional wholesaler's fleet of vehicles, who distributes his daily product pallets of different goods. These journeys start either from a central warehouse or from rural areas and lead to various delivery points in and around a city, where the food orders are delivered. The underlying vehicle routing problem for the transport of food orders defines a time and resource planning problem where the most optimal schedule respecting time, space and capacity constraints is created: Each vehicle has a start time at its origin location and an end time at its target destination. For each food order the transport vehicle must be determined as well as its pickup time at its origin and its delivery time at its destination. The task is to select a sequence for visiting several locations in such a way that no location apart from the first is visited more than once. The whole journey should be at low-emission as possible and a round trip, i. e., the first stop must be the same as the last. Time windows for the collection or delivery of goods are taken into account as well as the loading capacities of the delivery vehicles used. Goods to be delivered first have to be loaded last. The emissions of the different vehicles in the fleet must

[1] Fraunhofer FOKUS, Kaiserin-Augusta Allee 31, 10589 Berlin, Deutschland armin.wolf@fokus.fraunhofer.de
[2] Fraunhofer FOKUS, Kaiserin-Augusta Allee 31, 10589 Berlin, Deutschland silke.cuno@fokus.fraunhofer.de

be taken into account. The routing service optimises these routes to reduce emissions and save fuel. The problem is characterised by the fact that in addition to the travel time between the stations to be served, the emissions for these journeys are also reflected on. Collecting emissions data is a considerable challenge, such that in most cases the driving distances multiplied by a vehicle-typical emission value (e. g., in [gCO_2equiv/km]) were used.[3]

2 Related Work

A state-of-the-art approximate solver for Travelling Salesperson Problems (TSPs) is the so-called Lin-Kernighan-Helsgaun (LKH) solver [He00; He09] implementing variable depth local search [LK73]. However, in vehicle routing solving the underlying TSP is not sufficient because additional constraints such as limited resources, time windows for pickup and delivery and precedence constraints must be satisfied, too. To overcome these limitations there is an extension of the LKH [He17] transforming such routing problems into standard symmetric travelling salesperson problems and handling constraints by means of penalty functions within the approximate local search. Beyond these approximate solution approaches, there are also exact solution approaches.

Recently, tour scheduling, similar to vehicle routing, was considered and solved based on Constraint Programming (CP) [DV21]. The main differences are that trailers are transported by autonomous trucks without consideration of load capacities nor consideration of target destinations of the trucks after a one-day period, which is, however, important for human drivers. Furthermore, the objective there is the reduction of empty tours while we focus on the minimisation of the emissions of greenhouse gases. Finally, a (mixed integer) linear solver is used in [DV21] while we use a finite integer domain constraint solver based on recent research results (cf. [Be12]). There filtering methods for the *WeightedCircuit* constraint are presented. This constraint and even the underlying filtering methods are highly important for modelling and solving round trip problems like the TSP or the vehicle routing problems considered here. The presented methods are based on a relaxed problem considering so-called *1-trees*. In [IR19; IR21b] filtering based on *k-cuts* is presented, where necessary conditions of the graph to be connected by a *Hamiltonian* cycle, i. e., a round trip, are considered to determine necessary mandatory edges or to remove edges not contained in any solution or to detect inconsistencies. A similar approach based on the idea of swapping k edges on a tour used within LKH solvers to improve (e. g., the distance or emission of) the tour is considered in the *k-opt* constraint [IR21a]. These filtering methods can be combined with the filtering on the *WeightedCircuit* constraint. However, their power depends on the "density"/"connectivity" of the considered graph. If the graph is strongly connected filtering will less effective. This can be the case when the upper bound of the total weight of the round trip resp. of the *Hamiltonian* cycle will be rather weak. Thus this kind of filtering is effective when the upper bound is close to the optimum such that the remaining graph is rather sparse.

[3] see also Sec. 5 on dealing with this challenge.

3 Emission-Reducing Vehicle Routing in Food Logistics

3.1 The Considered Routing Problem

We are looking for a solution to an *emission-reducing vehicle routing problem*, namely a schedule for a round trip of the vehicle starting at and ending at a depot, visiting all other destinations exactly once within their allowed time windows. This schedule consists of picking up and delivering the orders within the defined time windows, satisfying the required travel times between the locations while minimising the total greenhouse gas emission for the trip. Specifically, an *emission-reducing vehicle routing* problem is defined by

- A *depot* which is a place where the round trip starts and ends.
- A *vehicle* with a limited capacity, an earliest time and a latest time of availability.
- A list of *orders*, i. e., a list of goods to be delivered from an origin location, usually the depot, to another destination. Each order requires that
  - The collection and delivery are work tasks of a certain duration.
  - The goods must be collected from their origin within a given time window.
  - The goods are delivered to the destination within a given time window.
  - The goods are loaded into the vehicle in such a way that the last loaded goods are unloaded first, if possible.
  - The capacity required by the goods never exceeds the capacity of the vehicle.
- a *distance matrix* where the journey times between any two locations are given.
- an *emission matrix* giving the greenhouse gas emissions produced by the vehicle when driving between any two locations.

The main diagonals of both matrices are obviously zero and currently the matrices must be *symmetric*, i. e., the data for driving from A to B is the same as for driving from B to A.

3.2 Solution Approach and Innovation

Our vehicle routing problem is modelled and solved as a "Constraint Optimisation Problem" (COP). Emission-reducing vehicle routing is a specialisation of the "Travelling Salesperson Problem with Time Windows" (TSPTW) belonging to the class of NP-hard problems. Therefore Artificial Intelligence (AI) methods (in particular heuristic search methods) are used in combination with Operations Research (OR) filtering methods for exact solutions.[4]

[4] approximate solution approaches are referred in Section 2.

Thus, our solution approach is essentially based on a constraint optimisation model adopted from [DCP16]. For the *WeightedCircuit* constraint, we use graph-theoretical filtering methods adopted from [Be12] and further OR filtering methods, e. g., for the *AllDifferent* constraint the methods presented in [Ré94], to reduce the search space.

The implementation and application of recent research results and the deployment of a vehicle routing service that can help to reduce greenhouse gas emissions in food logistics which is available over the web using open standards is the main innovation of our approach.

3.3 Implementation and Deployment as Web Service

For emission-reducing vehicle routing we use the Java constraint solver library "firstCS" [Wo12]. This library is expanded to satisfy the application specific constraints in the problem model (cf. Section 3.2). Therefore, we implemented the according filtering methods. For improved vehicle routing schedules, a special heuristic search is implemented that determines loading and unloading activities for each vehicle mainly in a "last-in-first-out" order. In this way rearranging activities of the loads/pallets stored in the loading space during the delivery trip should generally be avoided. For optimisation we have chosen a monotonous "Branch & Bound" strategy. In this way, step-wise improved solutions, at least 5 % better than the previous solution, are found until there is not any better solution.

The emission-reducing vehicle routing was realised as a web service. It can be used by sending planning requests in a predefined JSON format via http(s) to the URL of the web service. Increasingly better planning results are then sent to the enquirer by e-mail, in JSON and CSV format, for his/her enquiry. Alternatively, using a uniquely generated and returned reference number of the request, the current best planning result can be queried via http as well. These asynchronous approaches were chosen because the calculation of round trips can be very time-consuming. An example of a planning request is given in Listing 1.

```json
{
    "timeUnit": "min",
    "capacityUnit": "palettes",
    "emissionUnit": "g (COO-equiv)",
    "locations": ["Berlin-Depot", ...,"Blankenfelde-Dahlewitzer"],
    "vehicles": [
        {
            "vehicle": "v#0815",
            "sourceLocation": "Berlin-Depot",
            "earliestDepartureTime": "2022-02-08T04:00:00Z",
            "targetLocation": "Berlin-Depot",
            "latestArrivalTime": "2022-02-08T13:00:00Z",
            "availableLoadCapacity": 33,
            "transitionTimes": [[0, ...29], ..., [29, ..., 0]],
            "transitionEmissions": [[0, ..., 14000], [14000, ..., 0]],
        }
    ],
```

```
    "orders": [
        {
            "order": "1170",
            "origination": "Berlin-Depot",
            "earliestPickupTime": "2022-02-08T01:00:00Z",
            "latestPickupTime": "2022-02-08T04:30:00Z",
            "destination": "Berlin-Drakestrasse",
            "earliestDeliveryTime": "2022-02-08T04:30:00Z",
            "latestDeliveryTime": "2022-02-08T13:00:00Z",
            "requiredLoadCapacity": 2,
            "loadingDuration": 1,
            "unloadingDuration": 1
        }, ...
    ]
}
```

List. 1: Sample planning request in JSON format

Such a JSON request (cf. List. 1) consists of data on physical dimensions, locations, technical data on the used vehicle (e. g., load capacity, availability, etc.), transition times and emissions of the vehicle for the route sections between the defined locations as well as data on the orders of a daily delivery tour. The durations for loading and unloading goods are ideally based on measurements; here they are empirical values. Loading must start within the earliest and latest pickup time, unloading within the earliest and latest delivery time. Using these data sets, planning requests could be made to optimise the round trips of the vehicles and the according pickup and delivery tasks.

4 First Results

First results are based on real-world data. There, the round trip and the schedule of the pickup and delivery tasks of the vehicle routing problem consist of 15 orders. This is presented in Table 1 where orders from a depot located in Berlin must be delivered to 15 different locations in and around Berlin. The journeys from the depot to the destinations and back to the depot produce a total emission of about $\approx$ 103.75 kg of greenhouse gas, which is minimal and about $\approx$ 10% less than the emission of the initial solution, which is about $\approx$ 116.5 kg of greenhouse gas. While the computation of the initial solution requires $\approx$ 60 secs.[5] the computation of the optimal solution takes about $\approx$ 150 secs. Similar results were obtained for another vehicle routing problem defined by 12 orders. There, the initial solution ($\approx$ 100.8 kg emission) is computed within $\approx$ 20 secs. and the optimal solution ($\approx$ 94.0 kg emission; $\approx$ 7% improvement) is computed within $\approx$ 21 secs. Table 1 contains for each order: its number, i. e., its identifier (first column), according tasks, i. e., pickup and delivery (second column), the locations where these tasks are performed (third column) and the earliest and latest start times of these tasks (last two columns). All tasks in Table 1 must be carried out in this chronological order. The differences between the earliest and latest start times define some buffer times, which become important when a task or a drive takes

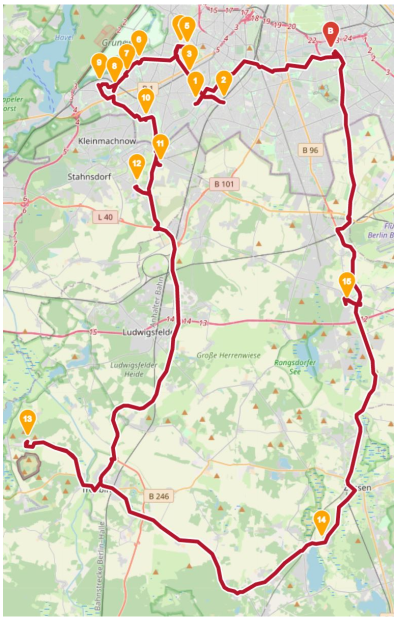
[5] Run-time measured on Intel Core i7-6600 CPU, 2.6 GHz, 12 GByte memory, Windows 10 Pro, OpenJDK 8u372.

longer than assumed. If there is enough buffer time then there will be no delay to the overall timetable and it is expected that all tasks can be completed within the given time slots.

[visualisation generated with `openrouteservice.org`]

Task schedule of a pickup-and-delivery round trip

order	task	location	earliest	latest
8517	pickup	Berlin-Depot	04:10	04:11
⋮	⋮	⋮	⋮	⋮
1158	pickup	Berlin-Depot	04:23	04:29
1430	pickup	Berlin-Depot	04:24	04:30
	depart.	Berlin-Depot	04:25	04:31
1430	delivery	Berlin-Ferdinandstraße	04:47	08:11
1158	delivery	Berlin-Glarner	04:55	08:19
1170	delivery	Berlin-Drakestraße	05:01	08:25
2645	delivery	Berlin-Königin-Luise-Str.	05:08	08:32
6198	delivery	Berlin-Hechtgraben	05:12	08:36
6353	delivery	Berlin-Onkel-Tom-Straße	05:22	08:46
6355	delivery	Berlin-Fischerhüttenstraße	05:26	08:50
8455	delivery	Berlin-Lindenthaler	05:31	08:55
1849	delivery	Berlin-Breisgauer	05:36	09:00
2790	delivery	Berlin-Claszeile	05:47	09:11
5576	delivery	Teltow-Kanada	05:57	09:21
8045	delivery	Teltow-Schenkendorfer	07:34	09:33
8517	delivery	Trebbin-Bismarckstraße	08:06	10:05
5045	delivery	Mellensee-Hauptstraße	08:37	10:36
3309	delivery	Blankenfelde-Dahlewitzer	09:02	11:01
	arrival	Berlin-Depot	09:32	13:00



Tab. 1: The task schedule and the optimal round trip of a vehicle routing problem with 15 orders

5 Conclusion, Discussion and Future Work

We developed a web service for emission-reducing vehicle routing in food logistics based on recent research results. First experiments show that the implemented functionalities meet the requirements. Next, there will be an evaluation in practice to check the feasibility of the routes and schedules as well as the validity of the data, e. g., are the journey times and emissions realistic, do they need to be adjusted?

Vehicle routing problems are inherently intractable, so we focus on problems faced by small and medium-sized enterprises to avoid long searches for solutions. In the future we aim to reduce the effort of collecting all data by using other services like `openrouteservice.org` for duration and distance matrices and use vehicle specific emission factors to compute the emissions from this received data if the real values are unknown. We also plan to generalise our solution approach so that *asymmetric* transition matrices can be considered, too. This is important, e. g., in hilly areas where emissions are different on routes in opposite directions (downhill vs. uphill). Furthermore, we like to examine whether the results in [IR21a; IR21b] or the alternative approach in [DCP16] can help to improve the performance of solution finding in our application context.

Acknowledgement

This work was carried out within the framework of the research and development project Stadt-Land-Fluss. The project is funded by the Federal Ministry of Food and Agriculture based on a resolution of the German Bundestag. Funding Reference Number: 2821KI002.

References

[Be12] Benchimol, P.; van Hoeve, W.-J.; Régin, J.-C.; Rousseau, L.-M.; Rueher, M.: Improved Filtering for Weighted Circuit Constraints. Constraints 17/3, pp. 205–233, July 2012, visited on: 08/09/2022.

[DCP16] Ducomman, S.; Cambazard, H.; Penz, B.: Alternative Filtering for the Weighted Circuit Constraint: Comparing Lower Bounds for the TSP and Solving TSPTW. Proceedings of the AAAI Conference on Artificial Intelligence 30/1, Mar. 2016, visited on: 08/09/2022.

[DV21] Dalmeijer, K.; Van Hentenryck, P.: Optimizing Freight Operations for Autonomous Transfer Hub Networks. arXiv:2110.12327 [math]/, Comment: 16 pages, 5 figures, 5 tables, Oct. 2021, arXiv: `2110.12327 [math]`, visited on: 01/25/2022.

[He00] Helsgaun, K.: An Effective Implementation of the Lin–Kernighan Traveling Salesman Heuristic. European Journal of Operational Research 126/1, pp. 106–130, Oct. 2000, visited on: 05/15/2023.

[He09] Helsgaun, K.: General K-Opt Submoves for the Lin–Kernighan TSP Heuristic. Mathematical Programming Computation 1/2, pp. 119–163, Oct. 2009, visited on: 05/15/2023.

[He17] Helsgaun, K.: An Extension of the Lin-Kernighan-Helsgaun TSP Solver for Constrained Traveling Salesman and Vehicle Routing Problems: Technical Report. In. Dec. 2017, visited on: 05/15/2023.

[IR19] Isoart, N.; Régin, J.-C.: Integration of Structural Constraints into TSP Models. In (Schiex, T.; de Givry, S., eds.): Principles and Practice of Constraint Programming. Lecture Notes in Computer Science, Springer International Publishing, Cham, pp. 284–299, 2019, ISBN: 978-3-030-30048-7.

[IR21a] Isoart, N.; Régin, J.-C.: A K-Opt Based Constraint for the TSP. In (Michel, L. D., ed.): 27th International Conference on Principles and Practice of Constraint Programming (CP 2021). Vol. 210. Leibniz International Proceedings in Informatics (LIPIcs), Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl, Germany, 30:1–30:16, 2021, ISBN: 978-3-95977-211-2, visited on: 08/10/2022.

[IR21b] Isoart, N.; Régin, J.-C.: A Linear Time Algorithm for the K-Cutset Constraint. In (Michel, L. D., ed.): 27th International Conference on Principles and Practice of Constraint Programming (CP 2021). Vol. 210. Leibniz International Proceedings in Informatics (LIPIcs), Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl, Germany, 29:1–29:16, 2021, ISBN: 978-3-95977-211-2, visited on: 05/18/2023.

[LK73] Lin, S.; Kernighan, B. W.: An Effective Heuristic Algorithm for the Traveling-Salesman Problem. Operations Research 21/2, pp. 498–516, 1973, JSTOR: 169020.

[Ré94] Régin, J.-C.: A Filtering Algorithm for Constraints of Difference in CSPs. In: AAAI Conference on Artificial Intelligence. [TLDR] A new filtering algorithm is presented that achieves the generalized arc-consistency condition for these non-binary constraints and has been successfully used in the system RESYN, to solve the subgraph isomorphism problem., Aug. 1994, visited on: 05/17/2023.

[Wo12] Wolf, A.: firstCS—New Aspects on Combining Constraint Programming with Object-Orientation in Java. en, KI - Künstliche Intelligenz 26/1, pp. 55–60, Feb. 2012.