

## Article

# Numerical Study on Planning Inductive Charging Infrastructures for Electric Service Vehicles on Airport Aprons

Niklas Pöch <sup>1,2,†</sup> , Inka Nozinski <sup>1,2,†</sup> , Justine Broihan <sup>1,†</sup>  and Stefan Helber <sup>1,\*,†</sup> <sup>1</sup> Department of Production Management, Leibniz University Hannover, 30167 Hannover, Germany<sup>2</sup> Cluster of Excellence SE<sup>2</sup>A—Sustainable and Energy-Efficient Aviation, Technische Universität Braunschweig, 38106 Braunschweig, Germany

\* Correspondence: stefan.helber@prod.uni-hannover.de; Tel.: +49-51-1762-5650

† These authors contributed equally to this work.

**Abstract:** Dynamic inductive charging is a contact-free technology to provide electric vehicles with energy while they are in motion, thus eliminating the need to conductively charge the batteries of those vehicles and, hence, the required vehicle downtimes. Airport aprons of commercial airports are potential systems to employ this charging technology to reduce aviation-induced CO<sub>2</sub> emissions. To date, many vehicles operating on airport aprons are equipped with internal combustion engines burning diesel fuel, hence contributing to CO<sub>2</sub> emissions and the global warming problem. However, airport aprons exhibit specific features that might make dynamic inductive charging technologies particularly interesting. It turns out that using this technology leads to some strategic infrastructure design questions for airport aprons about the spatial allocation of the required system components. In this paper, we experimentally analyze these design questions to explore under which conditions we can expect the resulting mathematical optimization problems to be relatively hard or easy to be solved, respectively, as well as the achievable solution quality. To this end, we report numerical results on a large-scale numerical study reflecting different types of spatial structures of terminals and airport aprons as they can be found at real-world airports.



**Citation:** Pöch, N.; Nozinski, I.; Broihan, J.; Helber, S. Numerical Study on Planning of Inductive Charging Infrastructures for Electric Service Vehicles on Airport Aprons. *Energies* **2022**, *15*, 6510. <https://doi.org/10.3390/en15186510>

Academic Editor: Adel El-Shahat

Received: 9 August 2022

Accepted: 5 September 2022

Published: 6 September 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

**Keywords:** dynamic wireless charging; electric vehicles; airport apron; airport infrastructure planning; electric busses

## 1. Introduction

A growing number of airports are electrifying their apron vehicle fleets to meet goals for climate-neutral airports (Bopst et al. [1], Interreg CENTRAL EUROPE [2], Flughafen München GmbH [3] and Royal Schiphol Group [4]). At Stuttgart Airport, for example, 40% of apron vehicles are equipped with electric drives and apron buses are already exclusively electrically powered (Bulach et al. [5]). Conductive charging is the state-of-the-art technology for charging these vehicles. However, this technology results in long downtimes due to vehicles charging and requires large batteries. A potential option for charging the vehicle batteries is dynamic inductive charging: Vehicles are wirelessly charged while in motion on a charging track installed below the road surface. This technology can substantially reduce downtimes. In addition, the need to have special charging stations is eliminated as well as the need for human involvement to plug in the charging cable. The objective of this paper is to report on methodological questions related to the potential usage of that charging technology for airport apron vehicles. We focus on the exemplary case of passenger buses transporting passengers from and to aircraft standing at outside parking positions. Still, we are convinced that the results hold for other types of service vehicles as well.

In order to charge apron vehicles with this technology, a dynamic inductive charging infrastructure would have to be implemented on the airport apron. This infrastructure consists of two components: the Power Supply Unit (PSU) and the Inductive Transmitter Unit (ITU). The PSU provides an alternating current of the required frequency to the ITU.

The ITU is installed below the road surface and charges the battery if a vehicle travels along (Panchal et al. [6]). Since the infrastructure requires high initial investments, only a fraction of the road network should be equipped with a charging track. At the same time, however, it must be spatially allocated in such a manner that the vehicles' batteries can be sufficiently charged while they are operating.

We use mathematical optimization models to formally characterize the problem of finding a spatial placement of the required infrastructure components such that the necessary capital investment is as small as possible. First models for planning inductive charging infrastructures on airport aprons have already been developed (Helber et al. [7] and Broihan et al. [8]). The standard approach to solving those models is to employ high-end commercial mixed-integer linear programming solvers such as Gurobi or CPLEX. However, Broihan et al. [8] have shown that solving real-world-sized test instances with standard solvers to proven mathematical optimality in a reasonable time is very often not possible.

This leads to the research questions addressed in this paper: Which features tend to make a particular instance of the infrastructure design problem of spatially allocating the components of the charging infrastructure on the airport apron road system hard to solve? Hard to solve in this context means that even within hours or days of computation time, it is not possible to find an infrastructure allocation that is known to be optimal in the mathematical sense of the underlying problem. If it turns out that this is indeed the case, a second question arises: Can we at least make a statement about the potential "optimality gap", i.e., indicating how far away from the optimal solution quality we can be at most? In order to answer these questions, we systematically generated a large-scale test bed of synthetic problem instances that reflect different types of real-world spatial structures of airport terminals as well as apron road networks and aircraft parking positions.

We will show that the proof of optimality takes a long time, although an admissible solution can already be found quickly. We will also investigate the influence of the problem's size and certain parameter specifications on the computation times. We show that the investments in the PSUs and ITUs, the vehicles' energy consumption and the energy intake can significantly impact the computation time.

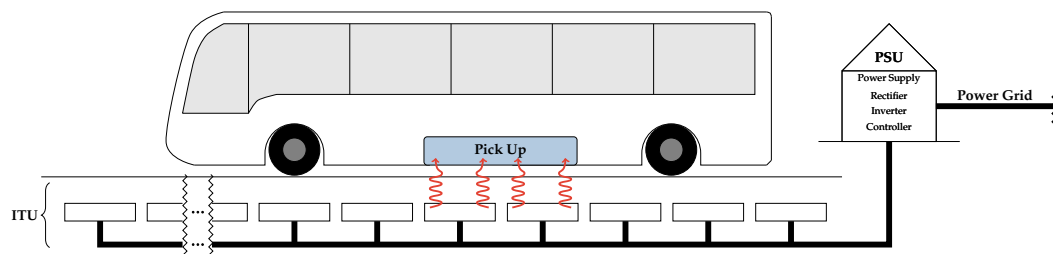
To this end, we analyze the properties of the Dynamic Inductive Charging Problem (DICP) experimentally to determine why this problem is difficult to solve for standard solvers. In particular, we examine the problem properties that lead to high computation times. The structure of the paper is such that we first provide a brief overview of the inductive charging technology, characterize the resulting airport apron design problem and report on related literature in Section 2. In Section 3.1, we formulate the model assumptions based on the previously stated properties of airport aprons. The introduced model in Section 3.2 is a variant of the optimization model presented in Broihan et al. [8]. Section 4 describes the generation of our instances that we use in the numerical study. We describe the general instance generation process in Section 4.2 and characterize the properties of the generated instance set for the analysis in Section 4.3. The results of the numerical study are presented in Section 5. We analyze the properties of the different instances and relate them to the computation time. Section 6 summarizes the results of this paper.

## 2. Characterization of the Problem Setting and Related Research

### 2.1. Dynamic Inductive Charging

Dynamic inductive charging means that the vehicle is charged wirelessly while in motion. For the wireless power transfer, primary coils are installed below the surface at selected elements of the airport apron road system (see Figure 1). Such a so-called ITU is supplied with an alternating current of the desired frequency by the PSU, which in turn needs a connection to the power grid. The PSU uses power electronics to modify the frequency. SAEJ2954 defines the frequency for wireless power transfer for electric vehicles in the range of 81.39–90 kHz (SAE International [9]). The electromagnetic field is created locally below the pickup unit of the moving vehicle, say, a passenger bus. Via the secondary coils within the vehicle's pickup unit, an alternating current is induced

by the electromagnetic field. This alternating current is rectified and used to charge the vehicle's battery and, eventually, power its electric engine. For further technical details on the technology of dynamic inductive charging and example projects, we refer to Li and Mi [10], Cirimele et al. [11], Lukic and Pantic [12], Panchal et al. [6], Ahmad et al. [13] and Covic and Boys [14].



**Figure 1.** Components of the dynamic inductive charging system (Source: Broihan et al. [8]).

The efficiency of the system is strongly dependent on the gap between the primary and secondary coil (Imura and Hori [15] and Moon et al. [16]). The smaller the gap between the primary and secondary coils, the higher the power transmission efficiency. For this reason, dynamic inductive charging systems are suitable for flat surfaces on which low-profiled vehicles operate. This is exactly the situation on airport aprons and one of the reasons why this charging technology might be interesting for airport aprons.

## 2.2. Energy Density, Trip Structures and Modeling of Battery Levels

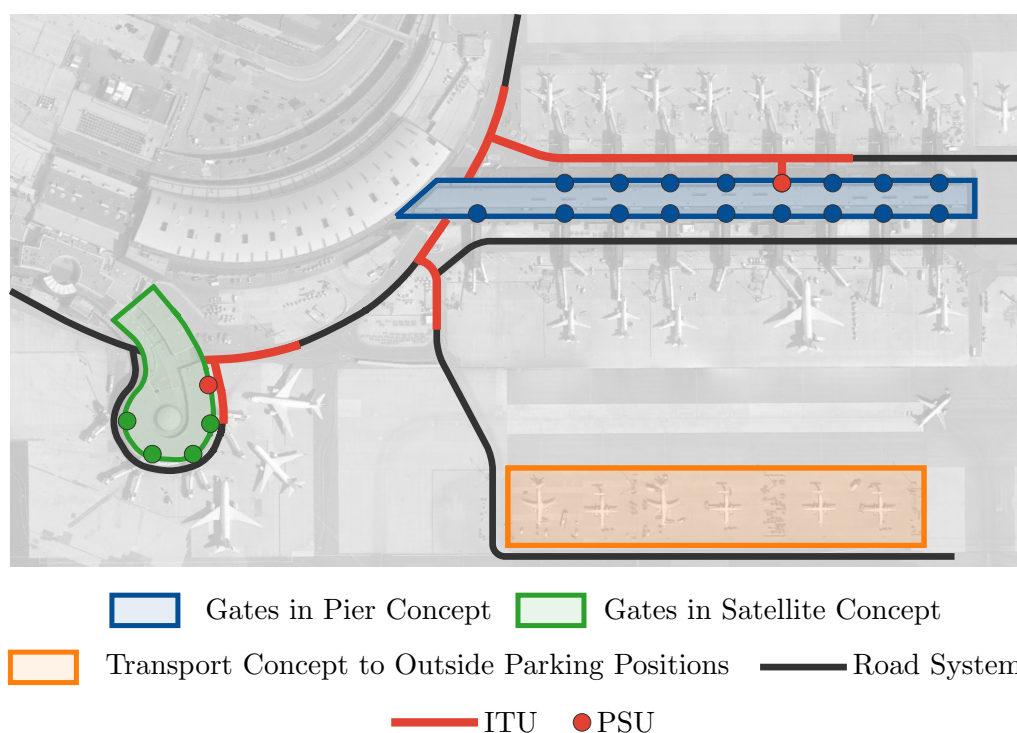
The energy density of electric batteries is known to be low relative to that of diesel fuel. Furthermore, the time required to transfer a certain amount of energy by charging its batteries, either conductively or inductively, is large compared to the time required to pump diesel fuel into the tank of a comparable vehicle with a combustion engine. Finally, an electric battery is not only expensive but also heavy due to its relatively low energy density. For all those reasons, the decision about the capacity of the battery of an electric vehicle is delicate from both the economic and the operational perspective: Very large batteries are not only expensive, but their transportation as part of the moving vehicle itself also consumes energy. On the other hand, small batteries require frequent re-charging and need to have ample spatially distributed charging facilities, again either for conductive or inductive charging.

For those reasons, many researchers studying dynamic inductive charging infrastructure design problems decided to model allocation decisions for ITUs together with battery size decisions for vehicles. The typical assumption is that a vehicle, say, a passenger bus serving an urban bus line, starts with a full battery at some initial location A, travels along a pre-defined route while serving a sequence of bus stations, and ends the tour at some final destination B. The charging infrastructure has to be allocated in such a way that the vehicle is never confronted with an empty battery while on its trip. To this end, the State of Charge (SOC) of the battery is tracked meticulously, considering both phases of de-charging and phases of charging (while passing ITU-equipped segments of the road system). An underlying assumption is that if the bus reaches the end of its tour, all that is needed is a battery that is not empty and that the battery will be fully charged before the bus begins its next trip. Examples of those modeling approaches can be found in Ko and Jang [17], Hwang et al. [18] and Ko et al. [19]. An important result of those studies is that the optimal structure of the charging infrastructure depends on the number of vehicles using it. Suppose only a small number of vehicles use the charging structure. In that case, it is beneficial to equip those few vehicles with large (and expensive) batteries to need only a few charging segments along the route the vehicles will travel. It is, however, not attractive to have a very large number of those vehicles equipped with large and expensive batteries. In this case, it is economically advisable to have a larger fraction of the road segments

equipped with ITUs and to be able to operate with smaller (and, hence, less costly) battery sizes in many vehicles.

When considering the question of how to allocate the ITUs and PSUs within the road network of an airport apron, it turns out that the spatial structure as well the nature of the trips driven by, say, passenger buses, differs substantially from those found in urban mass transportation by public buses.

Figure 2 presents as an example a selected part of Vienna Airport. On the airport apron, a road network connects the aircraft parking positions, equipment service areas, vehicle depots and terminal buildings adjacent to the airport apron. The parking positions for aircraft can be distinguished between gate positions and outside positions (Mensen [20]). Passengers can reach an aircraft parked at a gate position via a boarding bridge, while apron buses are used for transportation to the outside parking positions.



**Figure 2.** Selected part of the apron at Vienna Airport. Source: Vienna Airport [online], 48°07′03.39″ N 16°33′52.75″ E, Height 785 m, Google Earth © GeoBasis-DE/BKG 2009, URL: <http://www.google.com/earth> on 19 April 2022.

The layout of the terminal buildings determines the location of gate parking positions and the passenger gates. There are different terminal layout concepts, such as the pier, the satellite and the linear design. Figure 2 shows that these three layout concepts co-exist at Vienna airport.

At each terminal, several gates are available. Some gates are used exclusively to let passengers (un-)board their aircraft via a passenger bridge or to use passenger buses to transport the passengers to or from an outside aircraft parking position. In contrast, other gates can operate both with passenger bridges or passenger transportation buses.

During the turnaround of an aircraft, apron vehicles travel to and from the specific parking position, as well as gates or depots. A passenger bus, the exemplary type of vehicle considered in this paper, could pick up passengers at an aircraft and transport them to a terminal gate. Afterward, the bus could travel to another gate to pick up passengers and transport them from that gate to the parking position of their respective aircraft.

If we compare the operational elements of trips driven by passenger buses on airport aprons to those in urban public transport, we see three important differences:



- The distance and duration of a passenger bus trip on an airport apron are typically relatively short compared to those in public urban transportation.
- Most trips either start or end at one of the terminals or other dedicated airport areas.
- The number of different possible routes driven to serve the many possible service requests can be relatively large if we combine the different starting points, say, at the different terminal gates, with the many different destinations, for example at the outside aircraft parking positions.

Due to the third point, it seems impractical to follow the very detailed modeling approach introduced in Ko and Jang [17] and track at a very fine-grained resolution the SOC for the potentially extremely large number of conceivable service requests.

For this reason, we decided in this paper, as in Helber et al. [7] and Broihan et al. [8], to use a fundamentally different modeling approach. We require that the energy intake must be higher than the energy consumption for every service request. Therefore, the battery charge level at the end of a service request cannot be lower than at the beginning. As a result of that modeling decision, we do not need to model the SOC of the vehicles' battery in detail.

### 2.3. Modeling the Spatial Allocation of ITUs and PSUs

In Figure 2, we show in red a part of a fictitious allocation of ITUs and PSUs on some of the roads (in black) used by passenger buses for the Vienna Airport. Each ITU has to be part of a contiguous structure which is in turn connected to a PSU. Some locations, in particular those close to buildings, might be natural candidates to establish a PSU. On the other hand, there may be parts of the apron road network where installing ITUs could be impossible or unattractive due to interference with aircraft or the nature of road surfaces.

To represent the airport apron in the mathematical optimization model, we model the road network as a directed graph, as shown in Figure 3. Gates, parking positions and road intersections are modeled as nodes. The lanes of the airport apron roads are represented as links (directed arcs) in the graph. We assume that, on these links, the ITUs can be installed. If there are multiple adjacent edges, each with an installed ITU in the graph, they represent one contiguous ITU structure on the real airport apron. An example is given in Figure 3, where the ITU is installed between the nodes  $g_2$ ,  $i_6$ ,  $i_5$ ,  $i_{14}$  and  $i_{15}$ . This contiguous ITU structure can then be powered by one PSU, which is installed at node  $g_2$  in the example. A consistent connection from each ITU segment to a PSU is required. This connection can be set up directly if the ITU is directly adjacent to a PSU node or indirectly via other ITU segments (e.g., the ITU segment between  $i_5$  and  $i_6$  is connected via the ITU segment between  $g_2$  and  $i_6$  to the PSU at node  $g_2$ ).

We are now in the position to state in a non-technical manner the infrastructure allocation problem. The overall objective is to minimize the investment in ITUs and PSUs. The selection of links to be equipped with ITUs as well as the installation of PSUs must be such that:

- Each link equipped with an ITU is either connected directly to a PSU or indirectly via a neighboring ITU-equipped link on the shortest route to their respective common PSU;
- For each service request, the vehicle can take up at least as much energy while driving along the relevant links as it needs.

This modeling approach relieves us from the need to track the SOC of the vehicles' batteries. Figure 4 illustrates this. It shows an example SOC curve for a service request. The service request consists of the links  $l_1$  to  $l_6$ . At links  $l_2$  and  $l_5$ , an ITU is installed. When the vehicle passes over an ITU, it absorbs energy and the curve increases. In all other cases, the SOC decreases. According to our assumption, the vehicle must absorb at least as much energy as it consumes for the service request. For this reason, the SOC at the end might be greater than at the beginning. Of course, it may happen that the vehicle cannot absorb the energy because the battery is already full. In that case, the SOC can be lower at the end. We assume the battery is large enough to survive a longer part of the service request without energy intake. This results in the SOC never falling below zero.

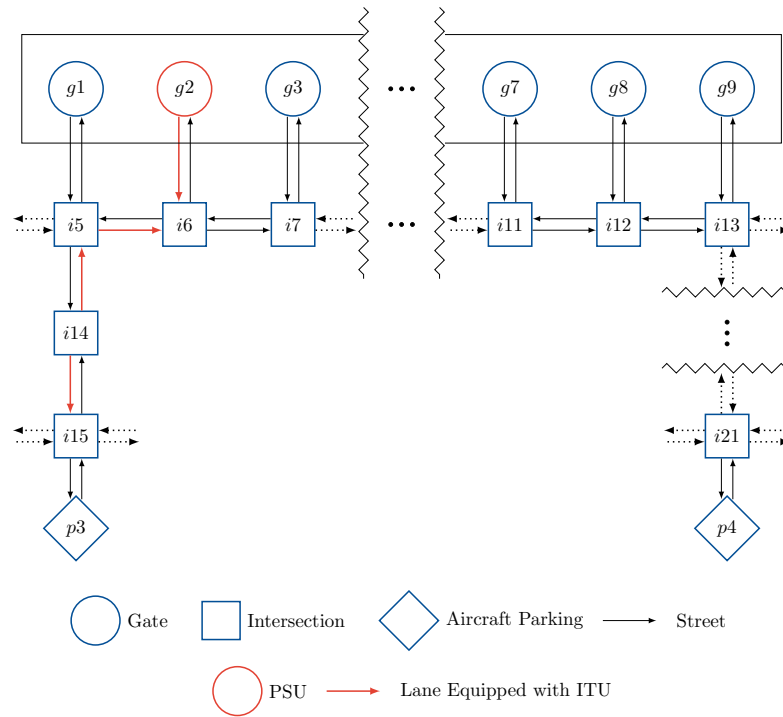


Figure 3. Representation of an airport as a directed graph. (Source: Adapted from Broihan et al. [8]).

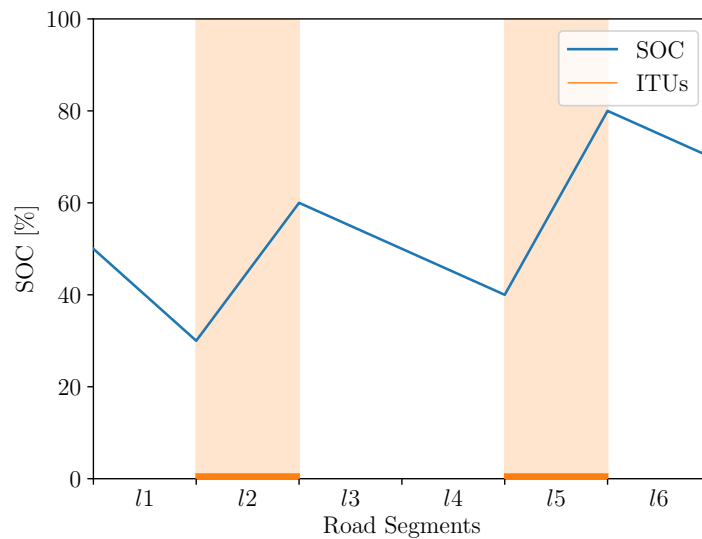


Figure 4. Exemplary SOC. (Source: Adapted from Broihan [21]).

2.4. Related Literature

For an overview of related literature on planning dynamic inductive charging infrastructure, we refer to Jang [22], Majhi et al. [23] and Yatnalkar and Narman [24]. They provide a comprehensive review of research articles and pilot projects. Most of these projects consider implementing a dynamic charging infrastructure for public bus systems. Many papers also consider battery capacity in the planning. Jang et al. [25] formulate a model in which the battery size is determined in addition to the placement of the charging infrastructure. The SOC is also considered in the model. The route is divided into segments that are either fully equipped with an ITU or not equipped at all. Ko and Jang [17], on the other hand, consider the route continuously. However, the model presented for this purpose is nonlinear. Both papers consider only one bus route at a time. In contrast, Hwang et al. [18] describe a multi-route environment. The model presented here has the special feature that the vehicles’ capacities can be different. Liu and Song [26] consider stochastic

elements in their model formulation for planning a charging infrastructure. They first present a deterministic model and then a model with uncertain travel time and energy consumption. The reason why particularly public buses are considered is that the tours are known in advance. The papers often consider so-called closed environments. That means systems where the influence of traffic and other external factors is small. In this case, the tour’s energy consumption and intake can be easily determined. Some papers consider other environments, e.g., Schwerdfeger et al. [27] looks at planning a charging infrastructure for highways.

Only Helber et al. [7] and Broihan et al. [8] considered the optimal placement of wireless charging infrastructures on airport aprons. Helber et al. [7] studied the characteristics of dynamic inductive charging on airport aprons and introduced the first mathematical optimization model for planning such an infrastructure. Broihan et al. [8] presented a reformulation and extension of this model considering multiple vehicle types and a service level restriction. In a numerical study, they analyzed test instances based on real airport aprons. For most of the instances, they could not prove optimality within a time limit of seven days with a standard solver. The resulting optimality gaps ranged from 8% to 15%. However, with the analyzed instances, they could not identify the reasons behind high computation times and the problem properties that lead to those. It is exactly this open question that we address in this paper.

### 3. The Dynamic Inductive Charging Problem

#### 3.1. Notation and Assumptions

To formally define the DICP, we now present the modeling assumptions as well as the notation summarized in Table 1. Note that assumptions and notation are closely related to those described in Helber et al. [7] and Broihan et al. [8], as the DICP is a generalization of the Dynamic Inductive Charging Problem considering Multiple Vehicle Types (DICP-MV) introduced in Broihan et al. [8].

**Table 1.** Notation used in the DICP model.

Indices and Sets	
$v \in \mathcal{V}$	vertices $\mathcal{V} := \{1, \dots, V\}$
$l \in \mathcal{L}$	links $\mathcal{L} := \{1, \dots, L\}$
$r \in \mathcal{R}$	service requests $\mathcal{R} := \{1, \dots, R\}$
$\mathcal{P} \subseteq \mathcal{V}$	set of vertices $v \in \mathcal{V}$ qualified to host a PSU
$\mathcal{P}_l \subseteq \mathcal{V}$	set of PSU candidates able to supply link $l \in \mathcal{L}$
$\mathcal{L}_v \subseteq \mathcal{L}$	set of links $l \in \mathcal{L}$ qualified to be powered by a PSU candidate at $v \in \mathcal{P}$
$\mathcal{L}_r \subseteq \mathcal{L}$	set of links $l \in \mathcal{L}$ included in service request $r \in \mathcal{R}$
$\mathcal{L}\mathcal{P}_v \subseteq \mathcal{L}$	set of links $l \in \mathcal{L}$ directly neighboring a PSU candidate at $v \in \mathcal{P}$
$\Gamma_{lv} \subseteq \mathcal{L}$	set of predecessors $l' \in \mathcal{L}$ of link $l \in \mathcal{L}$ in a direct connection on the shortest path to a PSU candidate at $v \in \mathcal{P}$
Parameters	
$c_v^{psu} \in \mathbb{R}^{\geq 0}$	investment in a PSU at vertex $v \in \mathcal{P}$
$c_l^{itu} \in \mathbb{R}^{\geq 0}$	investment in an ITU at link $l \in \mathcal{L}$
$e_{l1} \in \mathbb{R}^{\geq 0}$	energy intake by traveling on link $l \in \mathcal{L}$
$ec_r \in \mathbb{R}^{\geq 0}$	energy consumption by serving request $r \in \mathcal{R}$
Decision Variables	
$X_{lv} \in \{0, 1\}$	1, if link $l \in \mathcal{L}$ is equipped with an ITU and powered by PSU at vertex $v \in \mathcal{P}$ , 0 else
$Y_v \in \{0, 1\}$	1, if a PSU is installed at vertex $v \in \mathcal{P}$ , 0 else

- We model the road system of the airport apron as a directed and weighted planar connected digraph  $G(\mathcal{V}, \mathcal{L})$  composed of a set of vertices  $v \in \mathcal{V}$  and a set of directed edges or links  $l \in \mathcal{L}$  (see Figure 3).
- A vertex  $v \in \mathcal{V}$  defines a particular geographic location on the airport apron. This includes, for example, gate positions, aircraft parking positions, depots or fixed positions in the road system to split road sections.
- We define a subset  $\mathcal{P} \subseteq \mathcal{V}$ , which denotes the set of vertices  $v \in \mathcal{V}$  qualified to host a PSU. Installing a PSU at a candidate vertex  $v \in \mathcal{P}$  requires a fixed investment  $c_v^{psu} \in \mathbb{R}^{\geq 0}$ .
- Each directed link  $l \in \mathcal{L}$  represents a lane segment of the airport's road system. It connects a pair of adjacent vertices  $v \in \mathcal{V}$ . The direction of each link  $l \in \mathcal{L}$  corresponds to the direction of travel on this lane segment.
- We assume every link  $l \in \mathcal{L}$  to be a candidate to host an ITU. To supply such an ITU with electricity, it must be connected to a PSU, either directly or indirectly, via some other link. We define a set of links  $\mathcal{L}_v \subset \mathcal{L}$  which are qualified to be powered from a PSU installed at vertex  $v \in \mathcal{P}$ .
- Similarly, we define  $\mathcal{P}_l \subseteq \mathcal{V}$  as the set of vertices that could be equipped with a PSU powering link  $l$ .
- To ensure a physical connection from each ITU to its power-supplying PSU, we define a set of links  $l' \in \Gamma_{lv} \subset \mathcal{L}$  for each combination of link  $l$  and PSU candidate  $v$  as  $\Gamma_{lv} = \{l' \in \mathcal{L}_v \mid l' \text{ precedes } l \text{ on a shortest path to } v \in \mathcal{P}\}$ .
- $\mathcal{LP}_v \subset \mathcal{L}$  defines a set of links  $l$  directly neighboring a PSU candidate  $v \in \mathcal{P}$ .
- The installation of an ITU at link  $l \in \mathcal{L}$  requires a fixed investment denoted by  $c_l^{itu}$ .
- We define service requests  $r \in \mathcal{R}$  that represent the apron vehicles' potential service tasks, e.g., passenger transfers from a gate to an aircraft or baggage transportation.
- Serving request  $r \in \mathcal{R}$  requires the vehicle to move along a set of links  $l \in \mathcal{L}_r$ .
- We denote the consumed energy for a particular request  $r \in \mathcal{R}$  by traveling along the links in  $\mathcal{L}_r \subseteq \mathcal{L}$  by  $ec_r$ .
- As the vehicle serving request  $r$  travels along an ITU-equipped link  $\mathcal{L}_r \subseteq \mathcal{L}$ , it can take up to  $ei_l$  units of energy and charge its battery.
- For each request  $r$ , the sum of the potential energy intakes  $ei_l$  by traveling along the links in  $\mathcal{L}_r \subseteq \mathcal{L}$  must be at least as large as the energy consumption  $ec_r$  for that request. This assumption relieves us from the need to model the battery's SOC.

To describe the arrangement of charging infrastructure across the airport apron, we introduce two binary decision variables. The variable  $Y_v \in \{0, 1\}$  takes a value of 1 if a PSU is installed at the node  $v \in \mathcal{P}$  and 0 otherwise. Likewise,  $X_{lv} \in \{0, 1\}$  equals 1 if an ITU is installed at link  $l \in \mathcal{L}_v$  and connected to a PSU at vertex  $v \in \mathcal{P}$  and 0 otherwise.

### 3.2. Model Description

Based on the previous assumptions, we introduce the DICP, which is a generalization of the DICP-MV presented in Broihan et al. [8], as a linear program in binary variables as follows:

$$\min F = \sum_{v \in \mathcal{P}} \left( c_v^{psu} \cdot Y_v + \sum_{l \in \mathcal{L}_v} c_l^{itu} \cdot X_{lv} \right) \quad (1)$$

such that

$$\sum_{v \in \mathcal{P}} \sum_{l \in \mathcal{L}_r \cap \mathcal{L}_v} X_{lv} \cdot e_l \geq ec_r \quad r \in \mathcal{R} \quad (2)$$

$$X_{lv} \leq Y_v \quad v \in \mathcal{P}, l \in \mathcal{L}_v \quad (3)$$

$$\sum_{v \in \mathcal{P}_l} X_{lv} \leq 1 \quad l \in \mathcal{L} \quad (4)$$

$$\sum_{l' \in \Gamma_{lv}} X_{l'v} \geq X_{lv} \quad v \in \mathcal{P}, l \in \mathcal{L}_v \setminus \mathcal{LP}_v \quad (5)$$

$$X_{lv}, Y_v \in \{0, 1\} \quad v \in \mathcal{P}, l \in \mathcal{L}_v \quad (6)$$

The objective function (1) minimizes the total investment in the components of the dynamic-inductive charging infrastructure. Constraint (2) ensures a sufficiently large energy intake while serving a request  $r \in \mathcal{R}$ . According to constraint (3), ITU and PSU installation decisions are connected. Restriction (4) ensures that each link can be equipped at most once with an ITU, in which case it is connected to exactly one installed PSU. Restriction (5) enforces a connected ITU infrastructure from the furthest ITU along a shortest path to the powering PSU.

#### 4. Generating a Set of Instances for Dynamic Charging Infrastructures Problems

##### 4.1. Purpose and Objective of the Instance Generation Process

Our previous and preliminary numerical results (see Helber et al. [7] and Broihan et al. [8]) showed that any attempt to use a high-end commercial mixed-integer programming solver like Gurobi or CPLEX to solve the DICP shows very mixed results with respect to computation times and solution quality. In particular, we observed that when the instances

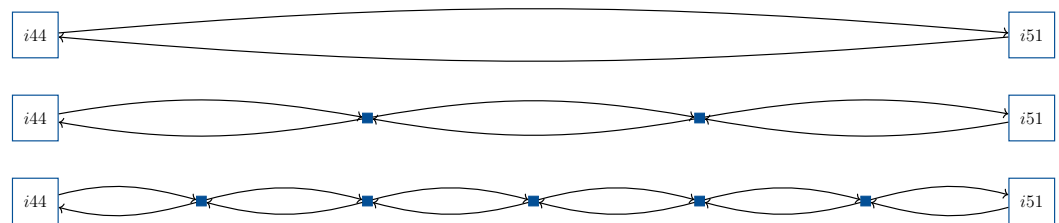
- tended to have a relatively small number of links  $l$  and vertices  $v$  in the graph representing the road network on the airport’s apron and, hence, also
- tended to have a relatively small set of service requests  $r$  to be considered,

is then those commercial solvers could often solve the resulting instances of the DICP to proven optimality within a few seconds or minutes.

However, real-world airports often have large and complex apron road networks, many passenger gates and many aircraft parking positions. As one consequence, the operational variance of the possible routings of the vehicles (passenger buses in our example) can be substantial. As we aim at obtaining a charging infrastructure allocation that is robust over a wide variety of such service requests, many of them have to be considered simultaneously in the infrastructure design decision, which is one factor leading to large model instances that tend to be hard to solve, i.e., having intolerably long computation times as well as potentially large optimality gaps.

A further problem of dealing with larger real-world airports is that the length of road segments, say, between terminals and aircraft parking positions, can be substantial. It could be desirable to equip only small fractions of those long road segments with the ITUs. However, to represent those fractions of the road segments in our model, we have to subdivide those road segments into sub-segments by adding additional vertices to the graph. An example of this problem aspect is depicted in Figure 5. Here, between the two nodes denoted as  $i44$  and  $i51$ , two long road segments exist, one for each direction and each having a length of 300 m. By introducing two or even five further vertices, link lengths of 100 m or even 50 m are created, respectively.





**Figure 5.** Extending an initial graph with link length maximum (**top**), 100 m (**mid**) and 50 m (**bottom**).

Having such finer granularity of the road system network in our model makes more cost-efficient infrastructure solutions possible, which is attractive. However, this comes at the price of operating again with larger models, i.e., models with a larger number of links and vertices that tend to be numerically more difficult to solve.

Another element that affects the difficulty of solving the problem by using standard commercial solvers to proven optimality is the cost ratio of the elements of the charging infrastructure, i.e., the necessary investment per PSU relative to the investment per ITU unit. If the PSUs tend to be relatively expensive, the resulting structures tend to operate with smaller numbers of PSUs to which then relatively large ITU structures are connected. In the opposite case of relatively inexpensive PSUs, larger numbers of ITU structures are placed over the road network.

Finally, the ITUs' power transfer also significantly impacts both the structure of the solutions and the difficulties of finding them. Suppose the ratio of the energy that can be picked up by a vehicle passing along a link is very large relative to the energy required to pass along that link. In that case, it may be possible to equip only a relatively small but well-chosen fraction of the apron road network with ITUs.

We know that all these factors affect:

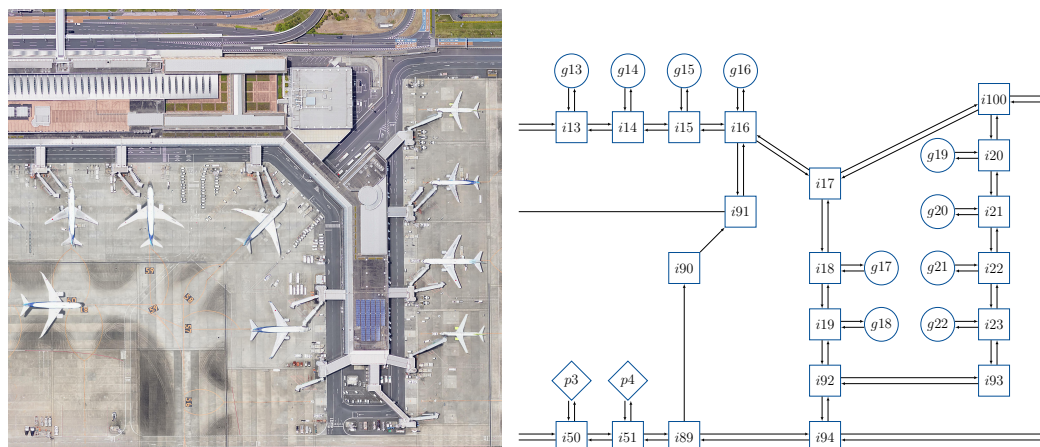
- The spatial structure of solutions to the infrastructure design problems;
- The computational time to find those solutions when using a standard solver such as Gurobi.

We also conjecture that there might be cross-effects between the influencing factors. In order to be able to identify those effects and to explore the limitations of using a commercial solver to solve the DICP, we systematically designed a full-factorial test bed consisting of hundreds of instances. We then solved those instances numerically to obtain experimental results shedding some light on the questions outlined above. Before we present the results of those computations, we first describe the underlying system of creating the test instances as well as their characteristic features.

#### 4.2. Design of the Instance Generation Process

For our numerical study, we need to be able to create many instances systematically. Therefore, we developed an instance generator that was implemented in Python 3.8. The procedure to generate instances can be divided into the three main steps of (i) creating a so-called initial graph, (ii) deriving from this initial graph a so-called instance graph, and (iii) adding further sets and parameters to arrive at a complete description of a planning instance.

In our case, the initial graph is created using the Python graph modeling package NetworkX and based on real airport apron structures. With the help of a satellite image, as presented in Figure 6 for a small part of Tokyo Airport Haneda, Japan, the graph nodes can be set according to the real airport's intersections, gates and parking positions. The weights of the edges correspond to the road segment's lengths of the real airport. An example of a complete initial graph is given in Figure 7; Hamburg Airport, Germany, inspires this one.



**Figure 6.** Comparison between excerpt of satellite image and excerpt of generated initial graph. Source: Tokyo Airport Haneda [online], 35°33′16.41″ N 139°47′17.42″ E, Height 6 m, Google Earth ©, URL: <http://www.google.com/earth> on 11 July 2022.

In the second step, we derived from the initial graph different instance graphs classes by deleting and adding nodes and edges to create instances that differ concerning the granularity of the modeled topology, resulting in instance classes denoted as “small”, “medium” and “large”. Graphs of the small instance class have only a small number of nodes and links. Thus, we must delete nodes and edges of the initial graph to adapt it to the small instance class. For the large instance class, we need to add nodes and edges to the initial graph. Within an instance class, the graphs derived from different initial graphs should have a comparable number of nodes and links. Since the initial graphs have different sizes, the number of links and nodes to be deleted or added differs for each initial graph. In the following, we will explain the mechanisms for deleting and adding nodes to create comparable instance graphs based on very different initial graphs.

Graph reduction is done by deleting gates, depots and parking positions and their associated intersections. For this purpose, the user specifies the portion of these positions to be deleted (deletion rate). The positions are then deleted at regular distances. Figure 8 shows an excerpt from the initial graph introduced in Figure 7 for different deletion rates. The figure at the top shows the result of a deletion rate of 0% (i.e., the initial graph), in the middle that of a deletion rate of 50%, and at the bottom the result of a deletion rate of 75%.

In addition to deleting nodes and edges, it is also possible and may be necessary to add them as explained in Section 4.1 and Figure 5 to achieve a finer granularity of the modeled topology and, hence, economically more efficient solutions.

In the third step, we generate the final instances for each instance graph. To this end, different parameters and sets are required, as indicated by the notation in Table 1. Table 2 summarizes how we derived the sets and parameters of the DICP. The set of links  $\mathcal{L}$  and nodes  $\mathcal{V}$  are directly taken from the instance graph. We consider all links to be candidates to host an ITU. Limiting the consideration to heavily trafficked sections of the road network would be possible. However, in this case, there is a risk that not all service requests can be served. This is particularly true if service requests do not use these route sections. For this reason, we consider all links as ITU candidates. We determine the PSU candidate positions  $\mathcal{P}$  from the set of all depots and gate positions to achieve a predetermined number of candidate positions so that those positions are evenly spread over the set of depots and gate positions. We assume that each PSU can supply each link. For this reason,  $\mathcal{L}_v$  equals  $\mathcal{L}$  for all PSU candidates. Conversely, this also means that every PSU can supply every link and, thus,  $\mathcal{P}_l$  equals  $\mathcal{P}$  for all links. Again, this is a simplified assumption. In reality, it would be conceivable that PSUs could not supply ITUs at any distance. This could be realized in our model if we consider in the set  $\mathcal{L}_v$  only the ITUs within a certain distance from the PSU  $v$ .

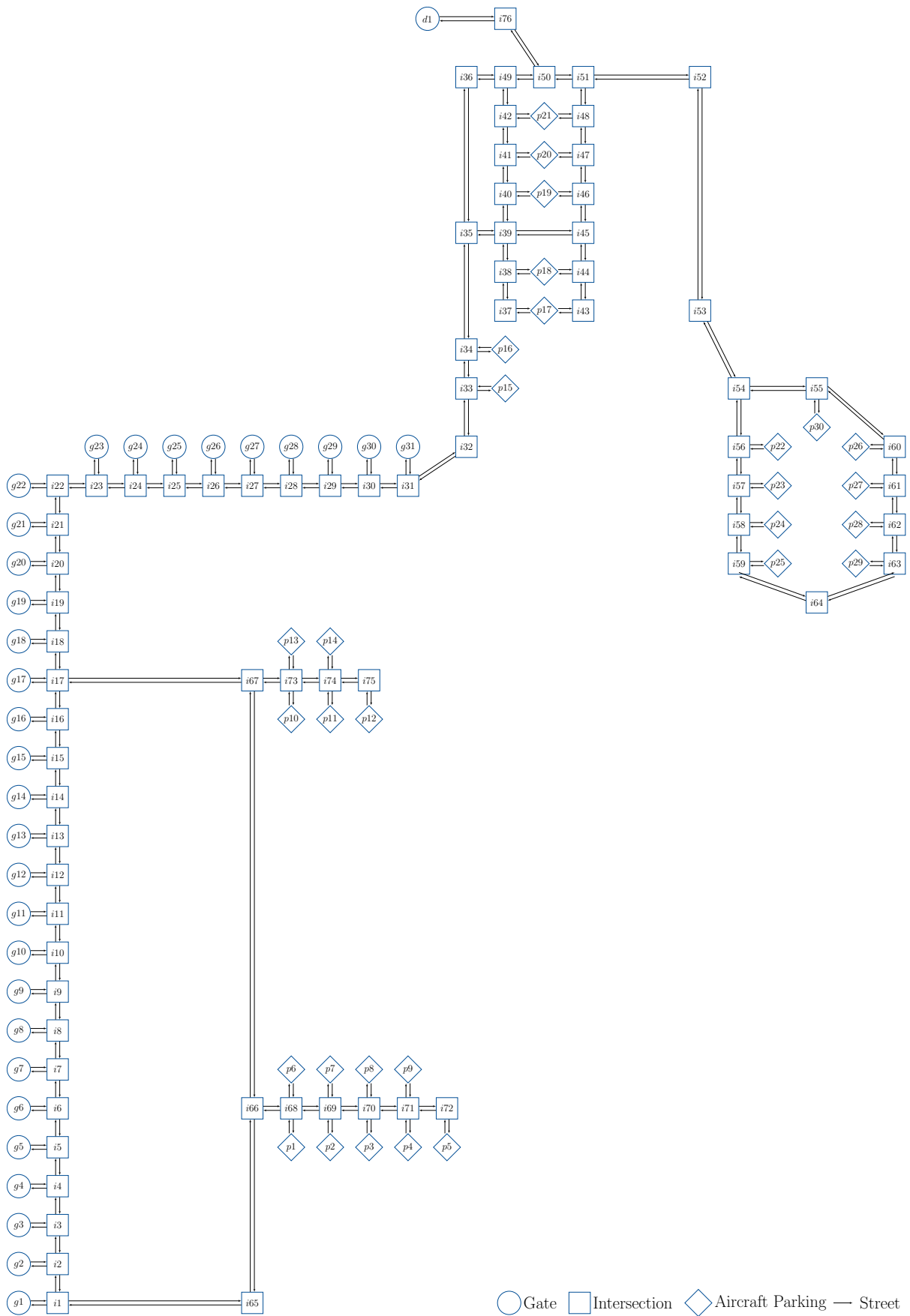
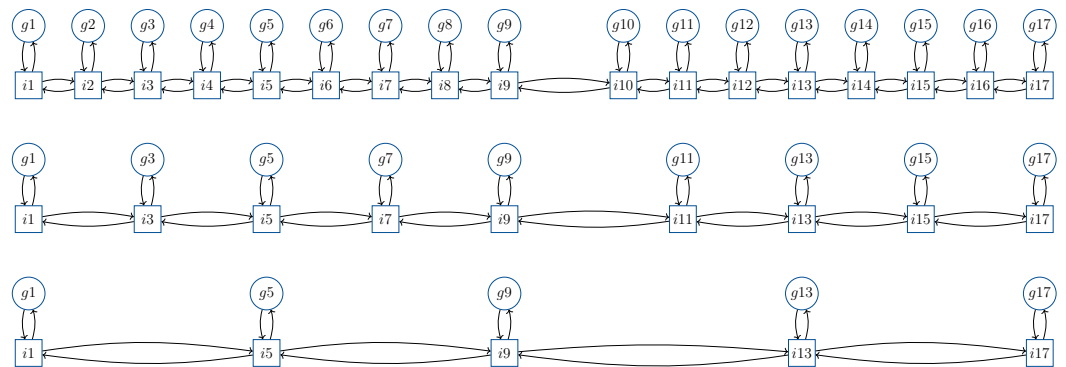


Figure 7. Example of an initial graph (Hamburg Airport).



**Figure 8.** Reducing an initial graph with deletion rate 0% (**top**), 50% (**mid**) and 75% (**bottom**).

Modeling the service requests is a crucial aspect of setting up problem instances. In all instances, we use the two-way request structure described in Broihan et al. [8]. Thus, a request starts at a terminal position (gate or depot), takes the shortest path to an aircraft parking position and ends again at a terminal position. All combinations of starting points (terminal positions), aircraft parking positions and endpoints (terminal positions) form the complete set of possible requests. Consequently, even in medium-sized instances, we get a very large set of requests. For the case of the initial graph in Figure 7, we have 31 passenger gates plus one bus depot and 30 outside aircraft parking positions. As a result, we have  $(31 + 1) \times 30 \times (31 + 1) = 30,720$  different two-way requests, leading to 30,720 constraints (2) in the DICP (1)–(6). For other (larger) airports the number of requests and corresponding model constraints may be substantially larger. However, these service requests often overlap, so it can be sufficient to consider only a part of them. Hence, we operate with different proportions of all possible requests. According to the given proportion, the requests for the set  $\mathcal{R}$  to be considered in the model are randomly selected from all possible requests for the given instance graph. We assume that the vehicles take the shortest path for each request and, hence, used shortest path algorithms to determine the set  $\mathcal{L}_r$  of links over which a vehicle travels as it serves request  $r$ . In addition, a set  $\Gamma_{lv}$  represents the predecessor of a link  $l$  on the shortest path to a PSU  $p$ . This set is determined using the Python package NetworkX in the preprocessing step to determine the shortest paths for all the requests.

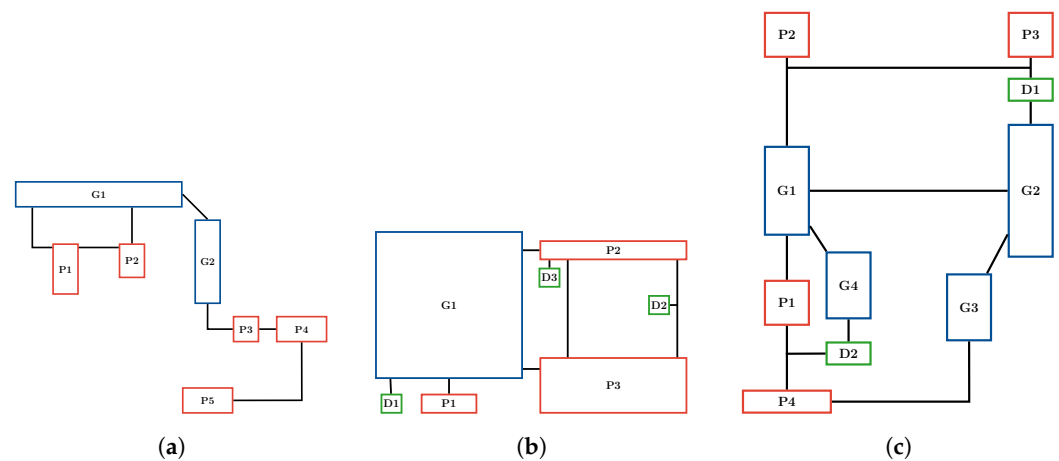
**Table 2.** Instance derivation from given data.

Input	Graph, energy intake per meter, energy consumption per meter, investment per PSU, investment in ITU per meter, percentage of requests, number of PSU candidates
Output:	$\mathcal{L}, \mathcal{V}, \mathcal{P}, \mathcal{L}_v, \mathcal{P}_l, \mathcal{R}, \mathcal{L}_r, \Gamma_{lv}, c_v^{psu}, c_l^{itu}, e_l, e_r$
Sets:	
$\mathcal{L}$	given by instance graph
$\mathcal{V}$	given by instance graph
$\mathcal{P}$	given number of PSU candidates is selected from graph
$\mathcal{L}_v$	equals $\mathcal{L}$ for all $v \in \mathcal{P}$
$\mathcal{P}_l$	equals $\mathcal{P}$ for all $l \in \mathcal{L}$
$\mathcal{R}$	from all given two-way request structures a given percentage of requests is chosen
$\mathcal{L}_r$	derived from graph
$\Gamma_{lv}$	derived from graph
Parameters:	
$c_v^{psu}$	equals given investment per PSU
$c_l^{itu}$	link length (derived from graph) is multiplied with given investment in ITU per meter
$e_l$	link length (derived from graph) is multiplied with given energy intake per meter
$e_r$	request length (derived from graph) is multiplied with given energy consumption per meter

We assume that the investment  $c_v^{psu}$  in a PSU at node  $v$  is equal for all  $v \in \mathcal{P}$  and is provided by the user. The investment  $c_l^{itu}$  to equip link  $l$  with an ITU is derived by multiplying the investment in ITU per meter with the link length. Similarly, the energy intake  $ei_l$  is determined by multiplying the energy intake per meter with the link length and the energy consumption  $ec_r$  by multiplying the energy consumption per meter with the request length. The length of the request is determined by adding the lengths of all links contained in this request.

#### 4.3. Description of the Generated Instances

For the numerical study conducted in this paper, we created a test instance set based on three initial graphs. These graphs are based on sections of real airport aprons but underwent minor modifications. Each graph in Figure 9 represents a certain apron layout. We refer to them as structures A, B and C. General layout A (left part of Figure 9) is an aggregate representation inspired by the conditions found at Hamburg airport, for which the initial graph was already introduced in Figure 7. The other two structures B and C (center and right part of Figure 9) were inspired by topologies found at other international airports, again without actually being isomorphic representations. Note that in these aggregate visualizations of Figure 9, an entry such as “G1” denotes an entire set of terminal gates in close proximity. Likewise, an entry “P1” represents an entire group of aircraft parking positions.



**Figure 9.** Aggregate Layout Structures. (a) Structure A. (b) Structure B. (c) Structure C.

We derived the initial graphs from satellite images. We set the positions for parking positions, terminal positions and intersections according to these images. Additionally, we placed intersection nodes at positions relevant to the structure (e.g., curves).

As mentioned before, we defined three instance classes of different sizes. Graphs of the instance class “small” have 55 to 65 nodes, graphs of the instance class “medium” have 100 to 120 nodes, and graphs of the instance class “large” have 180 to 200 nodes. We used different vertex (node) deletion rates and resulting link lengths to generate these instance graphs from each initial graph for structures A, B and C. The link length always specifies a maximum length since the links of the initial graph cannot always be divided without a remainder. Suppose we consider an instance graph with a link length of 100. In this case, a 250 m link of the initial graph will be divided into two links with a length of 100 m and one with a length of 50 m.

The values of deletion rates and link length are shown in Table 3. Excluding the initial graphs, this led to three different graphs per instance class and, hence, a total of nine different instance graphs.



**Table 3.** Graph Adjustments.

Instance Class	Structure	Nodes	Deletion Rate	Link Length
Base	A	138	0%	max.
	B	191	0%	max.
	C	293	0%	max.
Small	A	56	75%	max.
	B	57	85%	max.
	C	63	85%	max.
Medium	A	105	45%	100 m
	B	114	65%	200 m
	C	111	70%	200 m
Large	A	189	0%	50 m
	B	192	30%	100 m
	C	194	40%	100 m

In the numerical analysis, we analyzed the influence of selected sets and parameters on the computation time. Therefore, the following further attributes were varied when creating the instances:

- $|\mathcal{R}|$ : Proportion of considered service requests out of those potentially possible for the given instance graph;
- $|\mathcal{P}|$ : Number of PSU candidates;
- $c^{psu}/c^{itu}$ : Ratio of investment per PSU and investment in a meter of an ITU;
- $ei/ec$ : Ratio of energy intake per meter and energy consumption per meter.

By modifying the size of the set  $|\mathcal{P}|$  of nodes potentially hosting a PSU and the size  $|\mathcal{R}|$  of the set of service requests to consider, we varied the size of the problem. Therefore, we expected the computation time to increase for larger values for both attributes. However, we wanted to examine how significant the increase is. For the investment and energy parameters, only their ratios are relevant. As already explained, a high  $c^{psu}/c^{itu}$  ratio means that PSUs are expensive relative to ITUs. Consequently, few PSUs are expected to be built. Instead, more ITUs are built to connect all ITUs to the few installed PSUs. Thus, some ITU segments are built not because of the energy requirements of the vehicles but to produce a permissible (connected) charging infrastructure. A low  $c^{psu}/c^{itu}$  ratio can, on the contrary, lead to significantly more PSUs and fewer ITUs. With an  $ei/ec$  ratio close to one, energy intake and consumption are almost identical. As a result, nearly the entire infrastructure must be equipped with an ITU. An energy ratio  $ei/ec$  significantly larger than 1 indicates that the energy intake is significantly greater than the energy consumption. In such a case, only relatively few ITUs will be built. The influence of the parameters on the resulting infrastructure seems to be very clear. However, the influence on the computation time is not obvious, which is why we considered them in our analysis.

We considered three different values for every attribute, as shown in Table 4. The values were chosen to cover a wide range so that the influence of the attributes should be visible. For example, for the energy ratio  $ei/ec$ , a very low value (1.2) was selected, a value that may correspond to reality (3.24) and a particularly high value (10). We considered all different attribute combinations for all graphs. We have four attributes with three different values and thus have  $3^4 = 81$  instances for each graph of an instance class. Since we considered nine graphs, we obtained a total number of 729 instances. We solved these instances with Python 3.8 and Gurobi 9.1.0. Considering a strategic planning problem, we used a time limit for the computation of 48 h for each of the  $81 \times 9 = 729$  problem instances of our numerical test bed. In other words, we allowed up to about four years (!) of total Central Processing Unit (CPU) time for the entire study. All computations ran on the Dumbo subcluster of Leibniz University Hannover compute facilities, which uses an Intel

Xeon E5-4650v2 @2.40 GHz processor. Clearly, without such a cluster, the computations would not have been possible.

**Table 4.** Attributes of generated instances.

Attribute	Values
$ \mathcal{P} $	1, 4, 8
$\% \mathcal{R} $	30%, 70%, 100%
$c^{psu} / c^{itu}$	1, 25, 100
$ei / ec$	1.2, 3.24, 10

## 5. Numerical Results

### 5.1. Overview of Computation Times

Table 5 shows the average computation times ( $\bar{t}$ ), the median value ( $t^{median}$ ), the minimum computation time ( $t^{min}$ ) and the maximum computation time ( $t^{max}$ ) for each instance graph. Each row of the table gives the aggregate results over 81 different instances stemming from the systematic combination of the parameter entries in Table 4.

As expected, the computing time increases for larger instance classes, up to the maximum of 48 h allowed for the Gurobi solution process. Thus, the size of the network has a significant influence on the computation time. However, even within an instance class, the computing times vary remarkably. Structure C has significantly higher computing times in all instance classes, although the graph size is similar to the other structures. In addition, even within a instance graph, the computing times fluctuate enormously, as indicated by the wide range between minimum and maximum values.

Further, we observe that the median value is significantly smaller than the average for the small and medium instance classes. This is because, within an instance graph, many instances can be solved to proven optimality within seconds, minutes or a few hours, and some instances could not be solved in the given time limit (48 h). The outliers lead to high median values for these instance classes.

However, for large instances, the mean value is always smaller than the median value. This is because most instances could not be solved to optimality within the time limit and consequently have a reported computation time of 48 h.

**Table 5.** Computation Times.

		$\bar{t}$	$t^{median}$	$t^{min}$	$t^{max}$
small	A	142 s	14 s	1 s	2129 s
	B	204 s	57 s	1 s	4357 s
	C	719 s	155 s	1 s	10,322 s
medium	A	15 h	3 h	0 h	48 h
	B	10 h	1 h	0 h	48 h
	C	23 h	19 h	0 h	48 h
large	A	28 h	48 h	0 h	48 h
	B	34 h	48 h	0 h	48 h
	C	37 h	48 h	1 h	48 h

### 5.2. Analysis of Solution Process

Moreover, we studied the time needed by Gurobi to find the first feasible solution ( $\bar{t}^{first}$ ), the time after which the solution did not improve anymore ( $\bar{t}^{last}$ ), the size of the Linear Program (LP)-Gap ( $\overline{Gap}^{LP}$ ), and the size of the final Gap ( $\overline{Gap}^{final}$ ) when the time limit was reached. We calculated the LP-Gap as follows:  $Gap^{LP} = (OFV^{opt} - OFV^{rel}) / OFV^{rel}$ , where  $OFV^{opt}$  indicates the objective function value of the optimal solution and  $OFV^{rel}$  indicates the LP relaxation solution's objective function value. With this value, we want to

describe the quality of the LP relaxation. A low LP-Gap indicates a good LP relaxation and a high value indicates a bad LP relaxation.

Since we have a time limit of 48 h, optimality cannot be proven for all instances. Therefore, the final gap ( $\overline{Gap}^{final}$ ) indicates the gap between the upper and lower bound at the end of the time limit.

In addition, we performed a second numerical investigation in which the average number of equally optimal solutions ( $\bar{N}^{opt}$ ) was determined for the small instances. We consider solutions as equally optimal if they have the same objective function value as the optimal solution but differ in structure. Gurobi searches for all solutions with the same objective function value as the optimal solution and counts the number. Gurobi stops when all solutions are found or if the total number of equally optimal solutions is higher than 100,000. Due to the time limit, such a computation is not possible for the instances in the medium and large instance classes. Because of the upper limit, the average value of the equally optimal solutions  $\bar{N}^{opt}$  is underestimated.

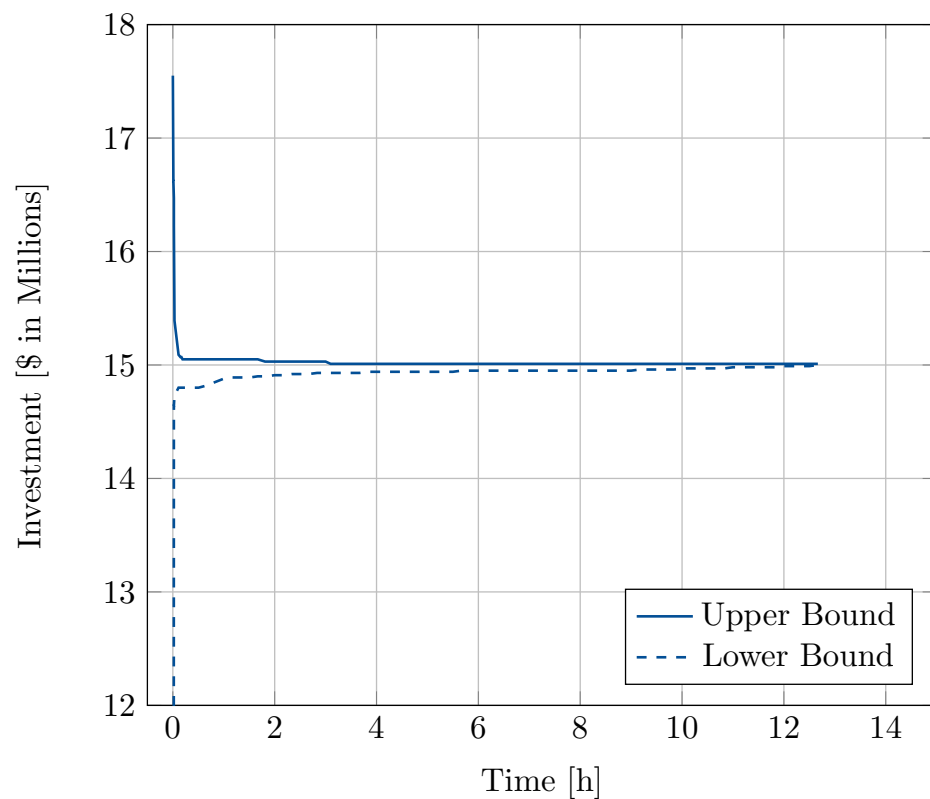
For all described metrics, the average values of 81 instances for each instance graph are shown in Table 6.

**Table 6.** Analysis of solution process.

		$\bar{t}$	$\bar{t}^{first}$	$\bar{t}^{last}$	$\overline{Gap}^{LP}$	$\overline{Gap}^{final}$	$\bar{N}^{opt}$
small	A	142 s	0 s	46 s	47%	0%	6251
	B	204 s	0 s	56 s	44%	0%	23,315
	C	719 s	0 s	256 s	56%	0%	26,525
medium	A	15 h	0 s	2 h	47%	0%	N/A
	B	10 h	0 s	1 h	57%	0%	N/A
	C	23 h	0 s	6 h	51%	1%	N/A
large	A	28 h	1 s	3 h	45%	4%	N/A
	B	34 h	0 s	9 h	48%	5%	N/A
	C	37 h	1 s	12 h	46%	7%	N/A

A first feasible solution is found on average in less than one second in each instance class. This shows that the problem's difficulty does not lie in finding a feasible solution. This is not surprising as a feasible (but expensive) solution can always be constructed by equipping each link with an ITU and then installing just one PSU.

Compared to the total computation time, a solution that is no longer improved is found after a relatively short time, particularly for small and medium-sized instances. Even for the hard-to-solve large instances, in most of the computation time, no improvement of the incumbent solution occurs. Figure 10 shows the upper and lower bound course stemming from the Branch-and-Bound process for one exemplary instance of a large-sized instance of structure A. We observe that a feasible solution is found very quickly. The upper bound, i.e., the objective function value of the best solution found so far, improves very strongly at first, then only very slightly. After about 3 h, the upper bound no longer improves. The lower bound slowly approaches the upper bound in the remaining ten hours until the optimization terminates with the proof of optimality of the Branch-and-Bound procedure used by Gurobi.



**Figure 10.** Solution Process of one Instance.

Table 6 shows that there are many equally optimal solutions for the instances of the small instance class. The high number of equally optimal solutions can be a reason for the long time needed to prove optimality. A large number of equally optimal solutions leads to many nodes to be visited in the Branch-and-Bound process, even if one optimal solution may already have been found. One reason for the high number of equivalent optimal solutions lies in the design of the instances. The links often have the same length and thus the same energy contributions and investments. Hence, many different infrastructure designs lead to the same energy contributions and investments. The structure also highly influences the number of equally optimal solutions and the computation time. If a structure is symmetrically built, many different infrastructures lead to the same energy contributions and investments. This is especially the case for structure C in our study. We observe the highest number of equally optimal solutions for this structure in the small instance class. But above all, we observe the highest computation times in all instance classes.

In addition, and making things worse, symmetric solutions occur, especially when the link length is reduced. Figure 11 shows an example of a symmetric solution. There is an intermediate node between nodes  $i1$  and  $i2$ . Let all links have the same length. In this case, both represented solutions are equivalent because they contribute the same energy to each service request and have the same investment. In any route segment that is not interrupted by an intersection and in which the link lengths are the same, each solution is equivalent in that the sum of the ITUs built in either direction is identical. In the example, the sum of ITUs in each direction equals 1. Consequently, with smaller link lengths, more symmetries are expected since there are significantly more such route sections.



**Figure 11.** Example of a symmetric solution.

As expected, the final gap increases with increasing problem difficulty. Again, we observe that instances of structure C are more difficult to solve than those of the other two structures. Larger instances mean that it is more likely that the time limit will be reached. For this reason, the final gap is larger with increasing problem size.

The mean value of the LP-Gap has similar sizes for all structures (between 40% and 60%). However, the LP-Gap varies extremely between the individual instances. As Table 7 shows, the most influential factor is the energy ratio. The table shows the average values of the LP-gap for all instances together with the respective energy ratio  $\frac{ei}{ec}$ . For an energy ratio of 1.2, the average LP-Gap is only 5%, for a ratio of 3.24 it increases to 26% and for a ratio of 10 even to 115%. For a high energy ratio, it is sufficient to only partially equip links with an ITU, especially if the links are very long. In the solution of the LP-relaxation of the DICP, these links are “utilized” with very small fractionals such as 0.1. Therefore, the LP-Gap is very poor (far away from the ideal of 0%). However, if the energy ratio is low, many ITUs must be built, and partial equipment is not useful. Thus the LP-Gap is smaller, reducing the numerical effort of the Branch-and-Bound process performed by Gurobi. Unfortunately, this numerically attractive situation is not very interesting from a practical point of view. It corresponds to a situation in which ITUs would need to be installed almost everywhere, which is exactly what one would like to avoid in the attempt to find an economically efficient dynamic charging infrastructure.

**Table 7.** Influence of energy ratio on the LP-Gap.

		$\overline{Gap}^{LP}$
	1.2	5%
$ei/ec$	3.24	26%
	10	115%

5.3. Influence of Problem Properties on the Computation Time

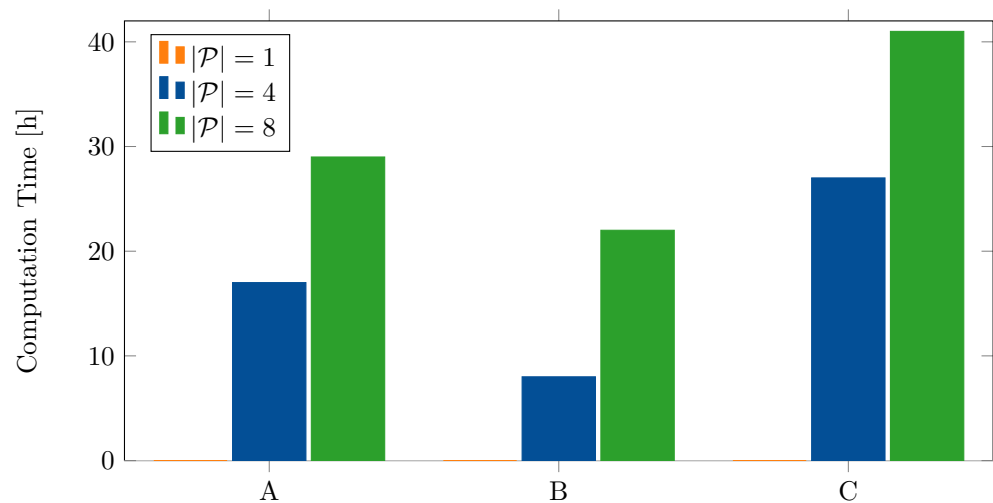
The influence of the different instance attributes from Table 4 on the computation time is shown in Table 8. The computation times are divided according to structure and the attributes and averaged over 27 values. In the first cell, for example, the times are averaged over all instances that belong to the small instance class of structure A and have only one PSU candidate. They differ in the number of service requests, the energy ratio and the ratio of investments ( $3 \times 3 \times 3 = 27$ ). In addition, the computation times of the medium-sized instances are presented graphically in Figures 12–15.

**Table 8.** Average computation times for different attributes.

		P			% R			$ei/ec$			$c^{psu}/c^{itu}$		
		1	4	8	30%	70%	100%	1.2	3.24	10	1	25	100
small (in min)	A	0	1	6	1	2	4	5	2	0	4	2	1
	B	0	1	9	4	2	4	6	3	1	6	2	2
	C	0	4	32	6	16	14	2	16	18	12	14	10
medium (in h)	A	0	17	29	13	16	17	23	21	3	20	19	7
	B	0	8	22	7	11	12	4	23	3	13	10	8
	C	0	27	41	22	23	23	24	32	11	25	24	19
large (in h)	A	1	37	46	25	29	30	21	33	30	32	26	25
	B	9	44	48	31	35	36	29	41	32	35	35	31
	C	14	48	48	36	37	38	35	42	33	38	36	36

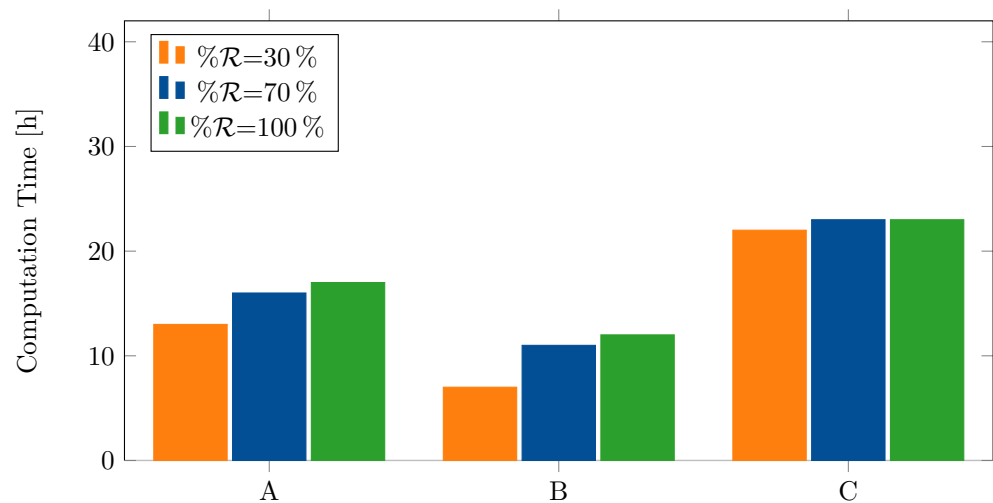
In Figure 12, we see that the number of PSU candidates strongly influences the computation time. The problem is easy to solve when only one PSU candidate is considered. In this case, no decision has to be made about where to place PSUs, but only where to place ITUs. This makes the problem much easier. Increasing the number of PSU candidates increases the computation time significantly.





**Figure 12.** Influence of the number of PSU candidates on the computation time for medium sized instances of structures A, B, and C.

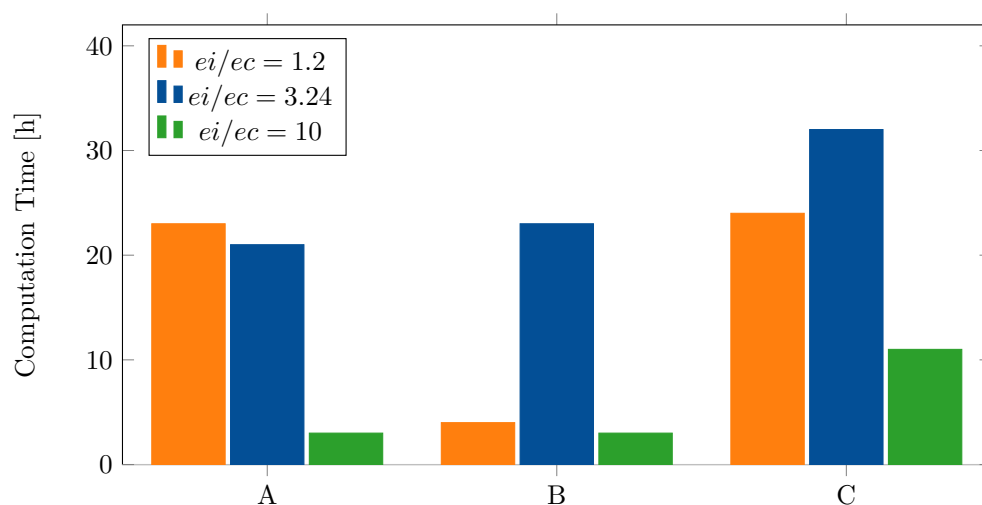
In Figure 13, we observe that increasing the number of service requests also leads to increased computation time. However, increasing the number of service requests leads to a much lower increase in the computation time than increasing the number of PSU candidates. On average, the increase from 30% to 70% is greater than the increase from 70% to 100%. The reason is the overlap of the requests. If all requests are included, an extremely high number of service requests is considered. Many of these requests have a high percentage of overlap. It is therefore conceivable that there are service requests that are already covered by the infrastructure requirements of other service requests. This means, for example, that if two service requests are fulfilled, a third request is automatically fulfilled as well. From the perspective of the problem, it makes no difference whether this third service request is considered or not. This effect will especially occur with a very large number of service requests. Thus, if we consider 70% of the requests, nearly all original requests are covered, and the difference between 100% is small. However, considering 30% of the requests will not cover all requests. Thus, the computation time increases substantially from 30% to 70% as more requests with no/few overlapping are added to the instance.



**Figure 13.** Influence of the percentage of considered requests on the computation time for large sized instances of structures A, B, and C.

Figure 14 shows the influence of the energy ratio on the computation time. For structure A of the medium instance, a ratio of 1.2 has the highest computation time, and for

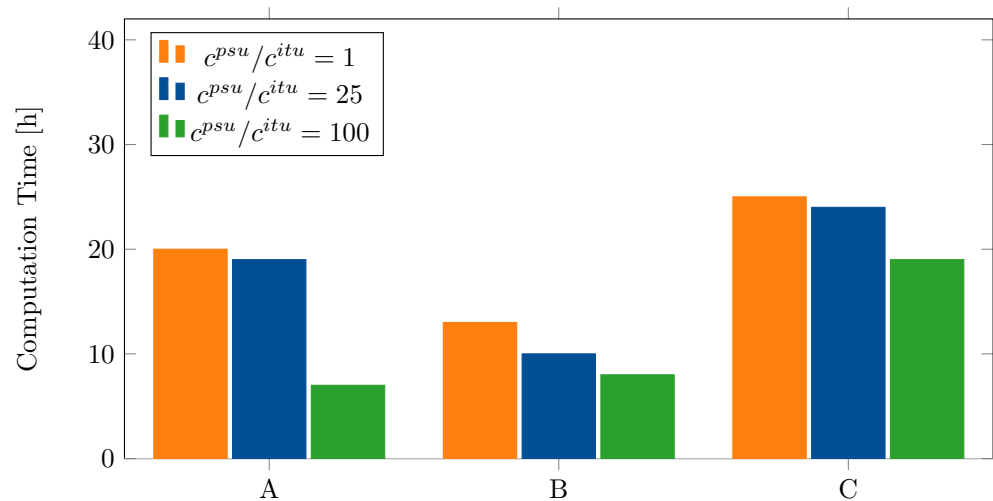
cases B and C, a ratio of 3.24. Table 8 also shows that the ratio of 3.24 mostly leads to the highest computation time. The combinatorics of the problem can explain this. An  $ei/ec$  ratio of 1.2 means that the energy intake is slightly larger than the energy consumption. Thus, almost the entire infrastructure has to be equipped with ITUs. There are few possibilities to realize this. The extreme case  $ei = ec$  illustrates this: In order to supply sufficient energy to the vehicles, the complete infrastructure must be equipped with ITUs. The only decision to be made is which PSU is to be connected. Therefore, there are only  $|P|$  different feasible solutions. If  $ei/ec$  is very large, then the energy intake is significantly greater than the energy consumption per distance unit of ITU passed by the vehicle. In this case, only a very small part of the infrastructure has to be equipped. For this reason, a very large part of the solutions can be excluded since they are too expensive. In the case of a medium-sized energy ratio, such as 3.24, many solutions are feasible and not too expensive, and many solutions can potentially be considered. For this reason, the computation time is usually the largest for this value.



**Figure 14.** Influence of the energy ratio on the computation time for large sized instances of structures A, B, and C.

Figure 15 shows that increasing the  $c^{psu}/c^{itu}$  ratio leads to lower computation times. This means a high investment in PSUs compared to ITUs lead to low computation times. As already shown, lower computing times occur if fewer PSUs are considered. If the PSUs are very expensive compared to the ITUs, it is not economical to build many of them. Rather, more ITUs would be built instead of one additional PSU. The question of how many PSUs should be built is not as difficult if the PSUs are very expensive. However, if the PSUs are less expensive, it may be useful to build multiple PSUs. In this case, there is the question of where to place the PSUs and whether it is better to build additional ITUs instead of one additional PSU. These aspects increase the combinatorial difficulty of the problem and thus the computing times.

In a final study, we re-ran the instances with the features that led to the highest computation times, but now with a CPU time limit of 200 h (as opposed to the initial 48 h). Table 9 shows the results. Optimality could not be proven for any of the three instances, even after 200 h of computation. On the other hand, the remaining integrality gaps between 3.7% and 8.4% indicate that even for those hard instances, the Gurobi solver was able to find solutions that may be acceptable from a practical point of view as their worst-case deviation from optimality is not too large.



**Figure 15.** Influence of the investment ratio on the computation time for medium sized instances of structures A, B, and C.

**Table 9.** Computation of most difficult instances.

		$ P $	$\% R $	$eilec$	$c^{psu}/c^{itu}$	CPU [h]	Gap
large	A	8	100%	3.24	1	200	3.7%
	B	8	100%	3.24	1	200	8.4%
	C	8	100%	3.24	1	200	7.3%

Finally, there is the question of what these computational results mean for real problem instances. In real problem instances, there will probably be many possible PSU locations because it is conceivable that there is a connection to the grid at every gate position. Modeling all of them would most likely make the problem intractable. However, several PSU locations close to each other could easily be represented by one PSU candidate. The justification for this approach is that we observed a very large number of equally optimal solutions, so it is non-unjustified to hope that we can drop many PSU candidate positions without losing all the optimal solutions from the solution space.

We further assume that there will be a large volume of traffic on the apron in the future. This is why we are looking at a large number of service requests. Instead of operating with a two-trip structure as in our instance generation process, we could also create instances with single-trip structures. This would reduce the number of requests to consider, making the problem numerically more tractable. On the other hand, it would lead to more infrastructure solutions that would be more expensive but more robust by giving the vehicles more charging opportunities.

As mentioned in Section 4, the energy ratio of 3.24 is based on actual data from Broihan et al. [8]. They assumed a power of 100 kW and efficiency of 80% for the wireless power transfer. Since this is still a new technology and the application is very specific, real data for the investments are difficult to find. Most applications consider a single bus. Jeong et al. [28] mentioned 500 \$/m as investment in a meter of an ITU and 50,000 \$/unit for a PSU. However, the PSU needed for this application cannot be compared to the application at the airport, where many vehicles are powered simultaneously. Nevertheless, we can summarize that real-world problems mostly have the characteristics that make the problem particularly difficult. In addition, we have made simplifications even in the graphs of the large instance class since we do not consider all gates, depots and parking positions. Furthermore, we have chosen a still very large link length. It is conceivable that there are savings with smaller link lengths. With instances of real problem size, we can therefore assume even significantly higher computation times.

## 6. Conclusions

This paper presents an optimization model for planning an inductive charging infrastructure on airport aprons. The model is a simplified variant of the model presented by [8]. The paper aimed to investigate the problem properties that lead to high computation times when solving the model with standard solvers and the reasons behind these high computation times.

For this purpose, we systematically generated a large set of instances in three steps. At first, we created initial graphs based on real airport apron structures. Afterward, we adapted these initial graphs to the three defined instance classes small, medium and large. Finally, we created instances from each graph with different characteristics, such as the ratio of energy intake to energy consumption.

In the numerical evaluation, we showed that current commercial standard solvers such as Gurobi can find a feasible solution quickly. They can even find good and potentially very good solutions in an acceptable time. However, proving the optimality of a solution often takes very long. One reason for this is the high number of equally optimal solutions, which are, among other reasons, due to symmetries in the problem structure.

The computation times of instances resulting from graphs of similar size vary significantly and depend on the instance's attributes. Specific attributes lead to remarkably high computation times. We showed that an increasing number of service requests and PSU candidates lead to higher computation times, whereby the influence of the PSU candidates is more significant. In addition, we investigated the influence of the ratio of energy intake to energy consumption and the ratio of PSU investment to ITU investment. The energy ratio significantly determines the size of the charging infrastructure. The numerical investigations showed that, in particular, an energy ratio that leads to a medium-sized charging infrastructure also leads to high computing times, as in this case, there tend to be many equally good alternatives. The investment ratio of PSUs to ITUs influences whether an additional PSU or more ITUs must be built. Comparatively cheap PSUs lead to high computing times.

In real applications, the problem instances may be even more complex than those considered in this paper. We showed that for some instances, optimality could not be proven by standard solvers within a time limit of eight days. For this reason, it is crucial to investigate how the computation time can be reduced. One possibility is to break symmetries by extending the model with symmetry-breaking restrictions or considering undirected graphs. This study showed that the presented rather intuitive model formulation is very hard to solve. For this reason, one might search for alternative model formulations that lead to lower computation times. For example, a flow-based model formulation in which the PSUs serve as sources and the ITUs serve as sinks would be conceivable. Further, future research projects should consider the use of heuristics and customized solution approaches.

Further research could further investigate the interaction of the power dimensioning and the allocation of the charging infrastructure's components. The model presented here hence provides substantial opportunities for such extensions.

**Author Contributions:** Conceptualization, N.P., I.N., J.B. and S.H.; methodology, N.P.; software, N.P. and J.B.; validation, N.P., I.N., J.B. and S.H.; formal analysis, N.P., I.N. and J.B.; investigation, N.P., I.N. and J.B.; data curation, N.P., I.N. and J.B.; writing—original draft preparation, N.P., I.N. and J.B.; writing—review and editing, S.H.; visualization, N.P., I.N. and J.B.; supervision, S.H.; project administration, S.H.; funding acquisition, S.H. All authors have read and agreed to the published version of the manuscript.

**Funding:** We would like to acknowledge the funding by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy—EXC 2163/1—Sustainable and Energy Efficient Aviation—Project-ID 390881007. The publication of this article was funded by the Open Access Fund of the Leibniz University Hannover.

**Data Availability Statement:** The data presented in this study are openly available in the Research Data Repository of the Leibniz University Hannover at <https://doi.org/10.25835/itr694hg> (accessed on 8 August 2022).

**Acknowledgments:** We acknowledge the support of the cluster system team at the Leibniz University Hannover, Germany, in the production of this work.

**Conflicts of Interest:** The authors declare no conflict of interest.

### Abbreviations

The following abbreviations are used in this manuscript:

CPU	Central Processing Unit
DICP	Dynamic Inductive Charging Problem
DICP-MV	Dynamic Inductive Charging Problem considering Multiple Vehicle Types
ITU	Inductive Transmitter Unit
LP	Linear Program
PSU	Power Supply Unit
SOC	State of Charge

### References

- Bopst, J.; Herbener, R.; Hölzer-Schopohl, O.; Lindmaier, J.; Myck, T.; Weiß, J. *Umweltschonender Luftverkehr*; Für Mensch & Umwelt: Dessau-Roßlau, Germany, 2019.
- Interreg CENTRAL EUROPE. *Electric Mobility Action Plan*; Interreg CENTRAL EUROPE: Vienna, Austria, 2019.
- Flughafen München GmbH. Electric Vehicles at the Airport. Available online: <https://www.munich-airport.com/electric-vehicles-at-the-airport-5938664> (accessed on 25 April 2022).
- Royal Schiphol Group. Electric Transportation between Gate and Plane. Available online: <https://www.schiphol.nl/en/schiphol-group/page/electric-transport-between-aircraft-and-gate/> (accessed on 25 April 2022).
- Bulach, W.; Hacker, F.; Haller, M.; Minnich, L.; Weber, M.; Hofmann, M.; Siehler, E.; Salzer, S. *Der Weg zur vollelektrischen Flughafenflotte—2. Working Paper aus dem Projekt Scale Up!*; Öko-Institut e.V.: Berlin, Germany, 2020.
- Panchal, C.; Stegen, S.; Lu, J. Review of static and dynamic wireless electric vehicle charging system. *Eng. Sci. Technol. Int. J.* **2018**, *21*, 922–937. doi: 10.1016/j.jestch.2018.06.015. [[CrossRef](#)]
- Helber, S.; Broihan, J.; Jang, Y.; Hecker, P.; Feuerle, T. Location Planning for Dynamic Wireless Charging Systems for Electric Airport Passenger Buses. *Energies* **2018**, *11*, 258. [[CrossRef](#)]
- Broihan, J.; Nozinski, L.; Pösch, N.; Helber, S. Designing Dynamic Inductive Charging Infrastructures for Airport Aprons with Multiple Vehicle Types. *Energies* **2022**, *15*, 4085. [[CrossRef](#)]
- SAE International. *Wireless Power Transfer for Light-Duty Plug-in/Electric Vehicles and Alignment Methodology*; SAE International: Warrendale, PA, USA, 26 August 2022.
- Li, S.; Mi, C.C. Wireless Power Transfer for Electric Vehicle Applications. *IEEE J. Emerg. Sel. Top. Power Electron.* **2015**, *3*, 4–17. [[CrossRef](#)]
- Cirimele, V.; Diana, M.; Freschi, F.; Mitolo, M. Inductive Power Transfer for Automotive Applications: State-of-the-Art and Future Trends. *IEEE Trans. Ind. Appl.* **2018**, *54*, 4069–4079. [[CrossRef](#)]
- Lukic, S.; Pantic, Z. Cutting the Cord: Static and Dynamic Inductive Wireless Charging of Electric Vehicles. *IEEE Electr. Mag.* **2013**, *1*, 57–64. [[CrossRef](#)]
- Ahmad, A.; Alam, M.S.; Chabaan, R. A Comprehensive Review of Wireless Charging Technologies for Electric Vehicles. *IEEE Trans. Transp. Electr.* **2018**, *4*, 38–63. [[CrossRef](#)]
- Covic, G.A.; Boys, J.T. Modern Trends in Inductive Power Transfer for Transportation Applications. *IEEE J. Emerg. Sel. Top. Power Electron.* **2013**, *1*, 28–41. [[CrossRef](#)]
- Imura, T.; Hori, Y. Maximizing Air Gap and Efficiency of Magnetic Resonant Coupling for Wireless Power Transfer Using Equivalent Circuit and Neumann Formula. *IEEE Trans. Ind. Electron.* **2011**, *58*, 4746–4752. [[CrossRef](#)]
- Moon, S.; Kim, B.C.; Cho, S.Y.; Ahn, C.H.; Moon, G.W. Analysis and Design of a Wireless Power Transfer System With an Intermediate Coil for High Efficiency. *IEEE Trans. Ind. Electron.* **2014**, *61*, 5861–5870. [[CrossRef](#)]
- Ko, Y.D.; Jang, Y.J. The Optimal System Design of the Online Electric Vehicle Utilizing Wireless Power Transmission Technology. *IEEE Trans. Intell. Transp. Syst.* **2013**, *14*, 1255–1265. [[CrossRef](#)]
- Hwang, I.; Jang, Y.J.; Ko, Y.D.; Lee, M.S. System Optimization for Dynamic Wireless Charging Electric Vehicles Operating in a Multiple-Route Environment. *IEEE Trans. Intell. Transp. Syst.* **2018**, *19*, 1709–1726. [[CrossRef](#)]
- Ko, Y.K.; Oh, Y.; Ryu, D.Y.; Ko, Y.D. Optimal Deployment of Wireless Charging Infrastructure for Electric Tram with Dual Operation Policy. *Vehicles* **2022**, *4*, 681–696. [[CrossRef](#)]
- Mensen, H. *Handbuch der Luftfahrt*; Springer: Berlin/Heidelberg, Germany, 2013. [[CrossRef](#)]

21. Broihan, J. Models and Algorithms for Designing Dynamic Inductive Charging Infrastructures on Airport Aprons. Unpublished Dissertation, 2022.
22. Jang, Y.J. Survey of the operation and system study on wireless charging electric vehicle systems. *Transp. Res. Part Emerg. Technol.* **2018**, *95*, 844–866. [[CrossRef](#)]
23. Majhi, R.C.; Ranjitkar, P.; Sheng, M.; Covic, G.A.; Wilson, D.J. A systematic review of charging infrastructure location problem for electric vehicles. *Transp. Rev.* **2021**, *41*, 432–455. [[CrossRef](#)]
24. Yalnalkar, G.; Narman, H. Survey on Wireless Charging and Placement of Stations for Electric Vehicles. In Proceedings of the 2018 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT), Louisville, KY, USA, 6–8 December 2018; pp. 526–531. [[CrossRef](#)]
25. Jang, Y.J.; Jeong, S.; Ko, Y.D. System optimization of the On-Line Electric Vehicle operating in a closed environment. *Comput. Ind. Eng.* **2015**, *80*, 222–235. [[CrossRef](#)]
26. Liu, Z.; Song, Z. Robust planning of dynamic wireless charging infrastructure for battery electric buses. *Transp. Res. Part Emerg. Technol.* **2017**, *83*, 77–103. [[CrossRef](#)]
27. Schwerdfeger, S.; Bock, S.; Boysen, N.; Briskorn, D. Optimizing the electrification of roads with charge-while-drive technology. *Eur. J. Oper. Res.* **2022**, *299*, 1111–1127. [[CrossRef](#)]
28. Jeong, S.; Jang, Y.J.; Kum, D. Economic Analysis of the Dynamic Charging Electric Vehicle. *IEEE Trans. Power Electron.* **2015**, *30*, 6368–6377. [[CrossRef](#)]