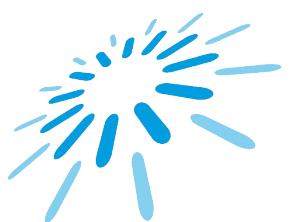


Algorithms and Data Structures

TH Aschaffenburg

Winter semester 2025/2026

Prof. Dr. Barbara Sprick



TH Aschaffenburg
university of applied sciences

Planning tool for autonomous drone delivery network in Nova Schilda

The city of **Nova Schilda** is planning to introduce an automated delivery system using autonomous drones. This system is intended to reliably transport packages between distribution centers, charging points, and end customers. To realize this goal, a **planning tool** is to be developed that centrally manages and optimizes routing, infrastructure, and resource allocation.

The city of Nova commissions **you** with the development of this planning system.

Functionality of the Tool

The planning tool for the drone network should offer the following functionalities:

B1: Import and visualize the drone network

B2: Define no-fly zones

B3: Extend or modify the drone network

F1: Check reachability

F2: Develop a method for determining efficient flight routes

F3: Compute delivery capacity

F4: Assess and improve network resilience

F5: Optimize placement of charging stations for large-scale coverage

F6: Set up a communications infrastructure for drones



Planning tool for autonomous drone delivery network in Nova Schilda

Task Description

Develop a [graph data structure](#) to represent, import/export, and modify the drone network, including charging points, distribution hubs, and delivery locations. ([Basic requirement](#))

Implement the specified [graph algorithms](#) (e.g., Dijkstra, Kruskal, Ford-Fulkerson/Edmonds-Karp, etc.). Use suitable test data (e.g., JSON file). ([Basic requirement](#))

Implement the described functionalities above. Clearly model each subproblem algorithmically.

(Use additional data structures such as heaps, queues, priority queues, or hash maps as needed.)

Programming Language: Java oder Python

Project Report

Submission Requirements (per team):

- The complete, well-documented source code (via GitLab)
- Input format for each function, including an example
- A [written report \(documentation\)](#) that includes, for each implemented function, an algorithmic modeling section covering the following points:
 - o What type of problem is being addressed?
 - o What modeling decisions were made?
 - o (e.g., directed/undirected graph, edge weights, states, additional data)
 - o Which algorithm was chosen, and why?
 - o Which data structures were used for your algorithm, and why?
 - o Were any alternative approaches considered?
 - o Complexity analysis of the key algorithms

Graph Input

You should be able to read the graph input from a **JSON-formatted file**.

Each **node** has an id and a type.

Each **edge** includes a start and end point plus weights if required.

An example file is provided on the right.

```
{  
  "nodes": [  
    {"id": "HUB", "type": "distributor"},  
    {"id": "A1", "type": "delivery"},  
    {"id": "A2", "type": "delivery"},  
    {"id": "CHARGE", "type": "charging"}  
  ],  
  
  "edges": [  
    {  
      "from": "HUB",  

```

Planning tool for autonomous drone delivery network in Nova Schilda

B1: Input drone network (from a JSON file)

This function reads in the drone network. The network should contain information about key nodes such as charging stations, distribution hubs, and delivery points, as well as the flight corridors connecting them.

A flight corridor may be usable in only one or in both directions, and energy consumption may differ by direction.

Corridors may also be capacity-limited (i.e., only a certain number of drones can use them simultaneously) or even blocked (e.g., temporarily restricted).

B2: Define up No-Fly zones (basic function)

This function allows the creation of temporary or permanent no-fly zones for flight corridors (e.g., due to disruptions or severe weather).

Specific corridors between two network nodes can be marked as restricted.

Optional Extension:

It is also possible to restrict an entire area defined by at least three coordinates, effectively creating a polygonal no-fly zone.

B3: extend and modify the drone network (basic function)

It should be possible to **extend the drone network** by adding both new **flight corridors** and new **nodes** (e.g., buildings or charging stations).

In some cases, the **energy cost** associated with specific corridors may also need to be adjusted.

Optional Extension:

Introduce a distinction between **safe**, **unsafe**, and **priority flight corridors**, for example by assigning different **weights or cost values** to reflect route preference or risk level.

Planning tool for autonomous drone delivery network in Nova Schilda

F1: Check Reachability

This function verifies whether all assigned **delivery points** in a given drone network are **reachable from a distribution hub**.

F2: Develop a Method for Determining Efficient Flight Routes

This function enables the calculation of **efficient single-delivery routes** for drones.

It determines the most efficient path from a **distribution hub** to a **delivery location**, taking into account both **energy consumption** and existing **no-fly zones**.

Optional Extension:

If a drone's energy capacity is insufficient for a direct delivery, it may **stop at a charging station en route** to recharge and continue the journey.

F3: Calculate Delivery Capacity

This function determines, for a given drone network, how many drones can **simultaneously (per hour)** deliver from a distribution hub to a specific **urban area** without exceeding the **capacity constraints of the flight corridors**.

An urban area typically includes **multiple delivery points**.



Planning tool for autonomous drone delivery network in Nova Schilda

F4: Assess and Improve Network Resilience

Resilience is crucial for the reliability of the drone network.

This function should enable the identification of critical connections within the network.

Specifically, it should help answer questions such as:

Which flight corridors represent bottlenecks?

What is the minimum set of corridors whose removal would split the network into disconnected parts?

F5: Optimize Charging Station Placement for Large-Scale Coverage (Optional)

Once the base network is in place, it should be optimized by adding additional charging stations to ensure drones can recharge during flight.

The goal is to place k charging stations in such a way that:

- The average distance to a charging station is minimized, and
- All delivery corridors are well-covered and accessible for recharging.

Planungstool für autonomes Drohnenliefersystem in der Stadt Nova Schilda

F6: Communication Infrastructure for Drones

During flight, drones rely on a **continuous connection** to a **central control unit** or to **each other** in order to: receive real-time delivery tasks, avoid collisions, and transmit updates or emergency signals.

To support this, the city plans to establish a **stable communication network** between all **fixed drone stations** using **directional radio links**.

The cost of setting up each link varies depending on **topography** and **distance**.

This function calculates a **communication network** that:

- connects **all drone stations**, and
- minimizes the **total setup cost**.

Planungstool für autonomes Drohnenliefersystem in der Stadt Nova Schilda

Die Stadt **Nova Schilda** plant die Einführung eines automatisierten Liefersystems mit autonomen Drohnen. Dieses System soll zuverlässig Pakete zwischen Verteilzentren, Ladepunkten und Endkunden transportieren. Um dieses Vorhaben zu realisieren, soll ein **Planungstool** entwickelt werden, das die zentrale Steuerung und Optimierung der Routen, Infrastruktur und Ressourcen übernimmt.

Die Stadt Nova beauftragt Sie mit der Entwicklung dieses Planungssystems.

Funktionalität des Tools

Das Planungstool für das Dronennetzwerk soll folgende Funktionalitäten bieten:

B1: Dronennetzwerk einlesen und darstellen

B2: Sperrzonen einrichten

B3: Dronennetzwerk erweitern oder anpassen

F1: Erreichbarkeit prüfen

F2: Entwicklung eines Verfahrens zur Bestimmung effizienter Flugrouten.

F3: Lieferkapazitäten

F4: Ausfallsicherheit prüfen und härten

F5: Ladepunktoptimierung für großflächige Versorgung

F6: Kommunikationsinfrastruktur für Drohnen



Planungstool für autonomes Drohnenliefersystem in der Stadt Nova Schilda

Aufgabenbeschreibung

Entwickeln Sie eine **Graphdatenstruktur** zur Repräsentation, Ein-/Ausgabe und Modifikation des Dronennetzes mit Ladepunkten, Verteilerhubs, Auslieferungspunkten. (Basis)

Implementieren Sie die genannten **Graphalgorithmen** (z. B. Dijkstra, Kruskal, Ford-Fulkerson/Edmonds-Karp, ...). Nutzen Sie geeignete Testdaten (JSON-Datei). (Basis)

Implementieren Sie die oben beschriebenen Funktionalitäten. Modellieren Sie dazu jedes Teilproblem klar algorithmisch.

(Nutzen Sie zusätzliche Datenstrukturen wie Heaps, Queues, Priority Queues oder HashMaps bei Bedarf.)

Programmiersprache: Java oder Python

Abgabe

Abgegeben werden soll pro Team:

- Der vollständige, kommentierte Quellcode (via GitLab)
- Eingabeformat für jede Funktion mit einem Beispiel

Die [Dokumentation](#) soll für jede Funktion eine **algorithmische Modellierung** enthalten:

- Welche Art von Problem liegt vor?
- Welche Modellierungsentscheidung wurde getroffen (gerichteter/ungerichteter Graph, Kosten, Zustände, Zusatzinformationen)?
- Welcher Algorithmus ist geeignet und warum?
- Welche Datenstruktur haben Sie für Ihren Algorithmus verwendet und warum?
- Welche Alternativen wurden ggf. in Betracht gezogen?
- Komplexitätsanalyse der wichtigsten Algorithmen

Eingabe der Graphen

- Alle Graphen sollen aus einer Datei im JSON Format eingelesen werden können. Rechts finden Sie eine Beispieldatei.
- Jeder Knoten hat eine id und einen Typ.
- Die Kanten haben einen Start und Endpunkt, sie können gerichtet oder ungerichtet sein, einen Energieverbrauch für die Drohnen angeben, eine Drohnenkapazität beinhalten oder auch eine Entfernung für eine Richtfunkstrecke.
- Mögliche Erweiterungen des Graphen wären „id“ für gezieltes Sperren von Flugkorridoren, Sperren („restricted“: true/false) oder Koordinaten („x“: 100, „y“: 50)

```
{  
  "nodes": [  
    {"id": "HUB", "type": "distributor"},  
    {"id": "A1", "type": "delivery"},  
    {"id": "A2", "type": "delivery"},  
    {"id": "CHARGE", "type": "charging"}  
  ],  
  
  "edges": [  
    {  
      "from": "HUB",  

```

Planungstool für autonomes Drohnenliefersystem in der Stadt Nova Schilda

B1: Dronennetzwerk einlesen (aus einer JSON Datei)

Diese Funktion liest das Dronennetzwerk ein. Das Netzwerk soll Informationen zu wichtigen Knotenpunkten wie Ladestationen und Verteilerhubs, Auslieferungspunkten, sowie zu den Flugkorridoren enthalten. Ein Flugkorridor kann in nur eine oder beide Richtungen benutzbar sein, der Energieverbrauch auf kann sich je Richtung unterscheiden. Auch können die Korridore kapazitätsbeschränkt (nur eine bestimmte Anzahl von Drohnen können ihn gleichzeitig benutzen) oder auch gesperrt sein.

B2: Sperrzonen einrichten (Basisfunktion)

Diese Funktion erlaubt das Einrichten von (temporären oder permanenten) Sperrzonen von Flugkorridoren (z. B. bei Störungen oder Unwetter). Hierfür können die Sperren für einzelnen Flugkorridore zwischen zwei Netzwerkknoten angegeben werden.

Erweiterung (optional)

Zusätzlich kann auch ein gesamtes Gebiet umgrenzt von mindestens 3 Koordinaten gesperrt werden.

B3: Dronennetzwerk erweitern und anpassen (Basisfunktion)

Es soll möglich sein, das Dronennetzwerk zu erweitern und sowohl neue Flugkorridore als auch neue Knotenpunkte (beispielsweise neue Gebäude oder Ladepunkte) hinzuzufügen. Manchmal muss für einzelne Korridore auch der Energielevel angepasst werden.

Planungstool für autonomes Drohnenliefersystem in der Stadt Nova Schilda

F1: Erreichbarkeit prüfen

Diese Funktion überprüft, ob in einem gegebenen Flugnetz von einem Verteilerhub aus alle zugeordneten Auslieferungspunkte erreichbar sind.

F2: Entwicklung eines Verfahrens zur Bestimmung effizienter Flugrouten.

Diese Funktion ermöglicht es, für die Drohnen effiziente Routen für Einzellieferungen zu berechnen. Was ist der effizienteste Weg für eine Drohne vom Verteilerhub zum Auslieferungsort? Hierbei sollen der Energieverbrauch aber auch Flugverbotszonen berücksichtigt werden.

Erweiterung (optional):

Reicht der Energieverbrauch einer Drohne nicht mehr aus, kann die Drohne auch unterwegs einen Ladepunkt anfliegen und wieder Energie aufnehmen.

F3: Lieferkapazitäten

Diese Funktion berechnet für ein gegebenes Flugnetzwerk, wie viele Drohnen gleichzeitig (pro Stunde) von einem Verteilerhub in ein bestimmtes Stadtgebiet liefern können, ohne die Korridorkapazitäten zu überschreiten. (Wobei ein Stadtgebiet typischerweise mehrere Auslieferungsorte umfasst.)

Planungstool für autonomes Drohnenliefersystem in der Stadt Nova Schilda

F4: Ausfallsicherheit prüfen und härten

Ausfallsicherheit ist für dieses Netzwerk von großer Bedeutung. Deswegen soll das Tool es ermöglichen, kritische Verbindungen zu identifizieren. (Welche Korridore bilden einen Flaschenhals? Welches ist die kleinste Menge an Korridoren, die das Netz in zwei Teile teilen könnte?)

F5: Ladepunktoptimierung für großflächige Versorgung (optional)

Nachdem das Grundnetz geplant ist, wollen Sie nun Ihr Netz optimieren und mehr Ladepunkte für die Drohnen einrichten, damit sie sich auf Ihrem Flug wieder mit Energie versorgen können. Platzieren Sie **k** **Ladepunkte** so im Netz, dass:

- die durchschnittliche Entfernung zu diesen Punkten minimiert wird,
- alle Lieferkorridore gut versorgt sind.



Planungstool für autonomes Drohnenliefersystem in der Stadt Nova Schilda

F6: Kommunikationsinfrastruktur für Drohnen

Die Drohnen sind während ihres Flugs auf eine durchgängige Verbindung zu einer zentralen Steuerinstanz oder untereinander angewiesen um Echtzeitaufträge zu empfangen, Kollisionen zu vermeiden oder Updates und Notrufe zu senden. Die Stadt möchte daher zwischen den fest eingerichteten Drohnenstationen ein stabiles Kommunikationsnetz durch Richtfunkstrecken aufbauen. Diese Einrichtung kostet je nach Topologie und Entfernung unterschiedlich viel. Diese Funktion berechnet ein Kommunikationsnetz, das alle Drohnenstationen miteinander verbindet, und gleichzeitig die Gesamtkosten minimiert.