



Hochschule für
Wirtschaft und Recht Berlin
Berlin School of Economics and Law

Optimaler Standort für einen Flugplatz

(Fallstudie #4)

Verfasser

Wang, Yanfei 77 211 875 289 s_wangy21@stud.hwr – berlin.de
Leonelli, Francesco 77 211 876 547 s_leonelli21@stud.hwr – berlin.de
Koenig, Konstantin 77 211 875 247 s_koenig21@stud.hwr – berlin.de

Vorlesung: Optimierung und Simulation (316502)

Dozent: Prof. Dr. Frank Brand

Semester: SS 2023

Abgabe: 19.06.2023

Inhaltsübersicht


| | |
|---|----|
| Eidesstattliche Erklärung | |
| 1 Executive Summary | 4 |
| 2 Einleitung | 5 |
| 3 Problemstellung | 6 |
| 4 Lösung | 7 |
| 4.1 Grafische Darstellung der Flugplätze A-F | 7 |
| 4.2 Die Qualitätsfunktion | 8 |
| 4.3 Die Nebenbedingungen | 9 |
| 4.3.1 Die Nebenbedingung I | 9 |
| 4.3.2 Die Nebenbedingung II | 10 |
| 4.3.3 Die Nebenbedingung I und Nebenbedingung II | 12 |
| 4.4 Der optimale Flugplatz G | 13 |
| 5 Sensitivitätsanalyse | 17 |
| 6 Mögliche Verallgemeinerungen des Problems | 19 |
| 6.1 Optimale Standorte zur Auslieferung von Paketen mit Flugdrohnen | 19 |
| 6.2 Optimale Standorte für Elektro-Ladestationen der BVG | 19 |
| 7 Schlussfolgerungen und Empfehlungen | 10 |
| 8 Zusammenfassung der Ergebnisse | 21 |

Eidesstattliche Erklärung

Hiermit erkläre ich an Eides Statt, dass ich die vorliegende Hausarbeit selbstständig und ohne fremde Hilfe verfasst und andere als die angegebenen Quellen und Hilfsmittel nicht benutzt habe. Die den benutzten Quellen wörtlich oder inhaltlich entnommenen Stellen (direkte oder indirekte Zitate) habe ich unter Benennung des Autors / der Autorin und der Fundstelle als solche kenntlich gemacht. Sollte ich die Arbeit anderweitig zu Prüfungszwecken eingereicht haben, sei es vollständig oder in Teilen, habe ich die Prüfer/innen und den Prüfungsausschuss hierüber informiert. Mir ist bekannt, dass falsche Angaben rechtliche Konsequenzen nach sich ziehen können.

Berlin/ 14.06.2023

Ort / Datum



Unterschrift Yanfei Wang



Unterschrift Francesco Leonelli



Unterschrift Konstantin König

1 Executive Summary

Die Luftfahrt gilt allgemein als großer Verursacher negativer Umwelteffekte. Dem Umweltbundesamt zufolge können rund 5 Prozent der weltweiten CO₂-Emissionen auf den globalen Flugverkehr zurückgeführt werden. Nachhaltigkeits- und Umwelterwägungen bei allen unternehmerischen Entscheidungen werden auch in der Luftfahrtbranche zunehmend wichtiger. Um fossile Ressourcen einzusparen, müssen Flugverbindungen und zurückgelegte Flugkilometer regelmäßig im Sinne der Nachhaltigkeit evaluiert werden.

Einen großen Beitrag hierbei kann die optimale Standortplanung für Flugplätze/-häfen leisten. Planungsverfahren sind in aller Regel kostspielig und langwierig und beziehen sowohl organisationelle, finanzielle, soziale als auch nachhaltige Aspekte mit ein.

Die folgende Arbeit widmet sich der Suche nach einem optimalen Flugplatz in Abhängigkeit zu einer bestehenden Flugplatzinfrastruktur. Ziel der Optimierung ist die Minimierung von zurückzulegenden Flugkilometern und somit der Einsparung von fossilen Ressourcen.

2 Einleitung

Die folgende Problemstellung ist ein weitverbreitetes Optimierungsproblem, welches allgemein unter dem Begriff **“Facility Location Problem (FLP)”** bekannt ist. Es gilt als NP-schwer und wurde maßgeblich durch den Wirtschaftswissenschaftler Alfred Weber untersucht.

(Vgl. Hanne, T., Dornbeger, R. (2023): Standortplanung und Netzwerkgestaltung, In: Computational Intelligence in Logistik und Supply Chain Management, Springer International Publishing, Seite 145)

Im Kern geht es darum die **optimale Position von Punkten oder Objekten (“facilities”)** innerhalb eines bestimmten Standortraums zu finden. Die optimale Position steht dabei in **Abhängigkeit zu Bedarfspunkten (“users”)**, welche den Einrichtungen zugeordnet werden. Es wird grundsätzlich davon ausgegangen, dass **Transaktionen** zwischen “facilities” und “users” **Kosten verursachen**. Zur Problemlösung wird daher häufig die Kostenfunktion minimiert sodass der optimale Standort als der **kostenminimalste Punkt** angesehen wird.

(Vgl. Barbati, M. (2021): Models and Algorithms for Facility Location Problems with equity considerations, S. 5; online unter: http://www.fedoa.unina.it/9474/1/Barbati_Maria_25.pdf [Zugriff am 14.06.2023])

Da die Standortplanung in der Realität als multikriterielles, komplexes Entscheidungsproblem gesehen wird, welches von verschiedenen Faktoren abhängt (Bodenbeschaffenheit, Baugenehmigungen, Sicherheitsabstände, etc.), wird versucht als Endergebnis eine “feasible region”, also einen eingeschränkten Standortraum zu bestimmen in dem optimale Punkte im Einklang mit allen Nebenbedingungen zu finden sind.

(Vgl. Hanne, T., Dornbeger, R. (2023): Standortplanung und Netzwerkgestaltung, In: Computational Intelligence in Logistik und Supply Chain Management, Springer International Publishing, Seite 129)

Praktische Anwendung findet dieses Optimierungsproblem im privaten als auch im öffentlichen Sektor bei der Suche nach dem optimalen Standort für Fabriken, Lagerhäusern, Verteilzentren, Krankenhäusern, Schulen, Versorgungszentren, Servern, Niederlassungen oder generell Anlaufpunkten wie z.B. der Gepäckausgabe an einem Flughafen. Die resultierenden Entscheidungen anhand der Optimierung sind meist **strategischer, langfristiger Natur und sind mit hohen Investitionskosten** verbunden.

3 Problemstellung

Die vorliegende Problemstellung umfasst die Bestimmung des **optimalen Standorts für einen Flugplatz**. Dieser neu zu definierende Flugplatz (Flugplatz G) steht in **Abhängigkeit zu sechs bereits bestehenden Flugplätzen** (Flugplatz A, B, C, D, E, F). Zukünftig sollen von Flugplatz G die anderen bereits bestehenden Flugplätze mit unterschiedlicher Intensität (z_i) angefliegen werden.

Gegeben sind die **x- sowie die y-Koordinaten für die Flugplätze A bis F** sowie **die Anzahl der Flüge (z_i)** zu den jeweiligen Flugplätzen. Darüber hinaus wird ein **geografisches Areal** ("Sumpfbiet") **durch eine Ungleichungsnebenbedingung ausgeschlossen**. Ein weiteres **geografisches Areal** ("See"), welches durch eine weitere **Ungleichungsnebenbedingung** ausgedrückt wird, darf ebenfalls nicht als Grundlage für den Standort verwendet werden.

Der optimale Standort soll mit Suchmethoden auf 10km genau bestimmt werden. Ziel der Optimierung soll derjenige Standort sein, **für den die Summe der täglich zurückzulegenden Flugkilometer (Luftlinie) minimal ist**.

```
In[1]:= zi = {40, 10, 20, 30, 20, 10};
coordinates = {{40, 200}, {160, 210}, {250, 160}, {220, 80}, {100, 40}, {30, 100}};
labels = {"A", "B", "C", "D", "E", "F"};
```

```
In[4]:= MatrixForm[
  {{Style["Flugplatz i", Bold], Style["x1 [km]", Bold], Style["y1 [km]", Bold],
    Style["Zahl der Flüge zi", Bold]}, {"A", 40, 200, 40}, {"B", 160, 210, 10},
    {"C", 250, 160, 20}, {"D", 220, 80, 30}, {"E", 100, 40, 20}, {"F", 30, 100, 10}}]
```

```
Out[4]//MatrixForm=
```

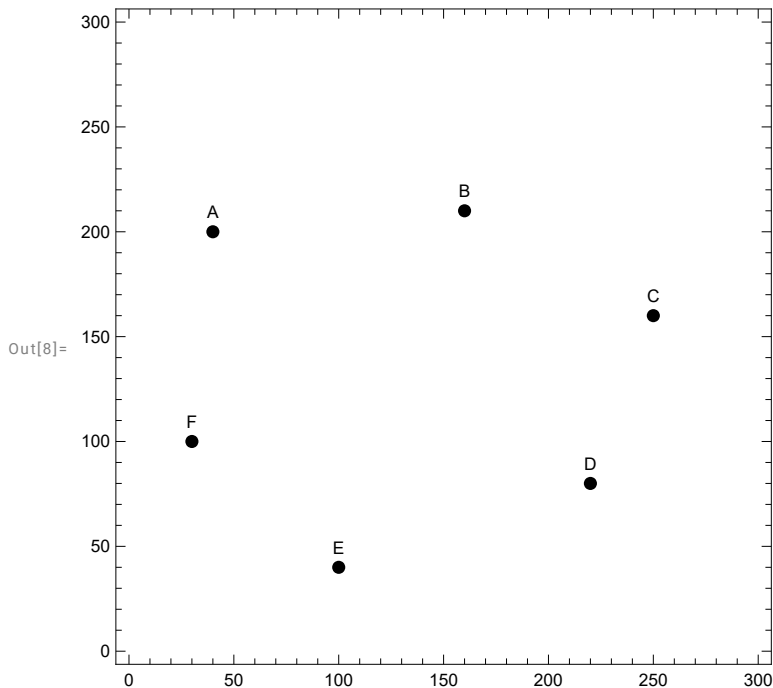
| Flugplatz i | x1 [km] | y1 [km] | Zahl der Flüge z _i |
|-------------|---------|---------|-------------------------------|
| A | 40 | 200 | 40 |
| B | 160 | 210 | 10 |
| C | 250 | 160 | 20 |
| D | 220 | 80 | 30 |
| E | 100 | 40 | 20 |
| F | 30 | 100 | 10 |

4 Lösung

4.1 Grafische Darstellung der Flugplätze A-F

```
In[5]:= zi = {40, 10, 20, 30, 20, 10};
coordinates = {{40, 200}, {160, 210}, {250, 160}, {220, 80}, {100, 40}, {30, 100}};
labels = {"A", "B", "C", "D", "E", "F"};
```

```
Show[ContourPlot[f == 0, {x, 0, 500}, {y, 0, 500}],
Graphics[PointSize[0.02], Point[coordinates]],
Graphics[MapThread[Text[#1, #2, {0, -2}] &, {labels, coordinates}]],
PlotRange -> {{0, 300}, {0, 300}}]
```



Anhand der xy-Koordinaten der Flugplätze A bis F ist ersichtlich, dass sich die Punkte innerhalb des ersten Quadranten des Koordinatensystems befinden. Die Visualisierung der Punkte sowie die Zielstellung eine Gesamtdistanz zu minimieren, grenzt die Suche nach der passenden Qualitätsfunktion ein.

4.2 Die Qualitätsfunktion

An allen Entscheidungsproblemen im Zusammenhang mit dieser Optimierungsaufgabe sind die Flugplatzstandorte, die Entfernungen zum neuen Standort G sowie die Anzahl der Flüge beteiligt. Jeder Flugplatz wird durch ein geordnetes Paar von Variablen repräsentiert, das den Koordinaten (x_i, y_i) in der Ausgangstabelle entspricht. Diese Koordinaten sind unveränderlich und entsprechen den Parametern des Optimierungsproblems, ebenso wie die Flugzahlen z_i . Die Flugzahlen z_i dienen als Gewichtung der Parameter in der Qualitätsfunktion.

Im Gegensatz dazu entsprechen die Koordinaten des neuen Flugplatzes G (x, y) den Variablen, die im Laufe des Lösungsprozesses des Problems festgelegt werden müssen. Da es sich um einen Distanzminimierungsvorgang handelt, könnte das Problem theoretisch mit einer euklidischen oder einer Manhattan-Distanz angegangen werden. Laut der Aufgabenstellung ist eine direkte Luftlinie als Distanz anzunehmen. Daher kann die Qualitätsfunktion bzw. Zielfunktion als euklidische Distanz wie folgt formuliert werden, entsprechend der Syntax, die im Unterricht verwendet wurde:

$$f(x, y / x_i, y_i, z_i) = \sum (z_i * \sqrt{(x - x_i)^2 + (y - y_i)^2}) \rightarrow \min$$

In Mathematica wurde diese Funktion wie folgt berechnet:

```
In[9]:= zi = {40, 10, 20, 30, 20, 10};
coordinates = {{40, 200}, {160, 210}, {250, 160}, {220, 80}, {100, 40}, {30, 100}};
labels = {"A", "B", "C", "D", "E", "F"};

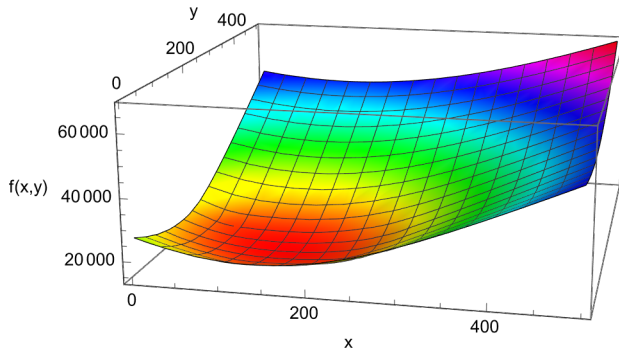
(* Objective function / Qualitätsfunktion*)
f[x_, y_] := Total[zi * Sqrt[(coordinates[[All, 1]] - x)^2 + (coordinates[[All, 2]] - y)^2]]
f[x_, y_]
```

```
Out[13]= 20 Sqrt[(100 - x_)^2 + (40 - y_)^2] + 30 Sqrt[(220 - x_)^2 + (80 - y_)^2] + 10 Sqrt[(30 - x_)^2 + (100 - y_)^2] +
20 Sqrt[(250 - x_)^2 + (160 - y_)^2] + 40 Sqrt[(40 - x_)^2 + (200 - y_)^2] + 10 Sqrt[(160 - x_)^2 + (210 - y_)^2]
```

Das Paar von Punkten (x, y) , das zur Minimierung des Problems führt, wird als optimale Lösung des Problems bezeichnet (Optimumsuche). Es war zu bestimmen, ob es sich um ein Problem der globalen Optimierung handelt, das heißt, ob die optimale Lösung des Problems mit einem globalen Minimum verbunden ist. Aus diesem Grund wurde nachfolgend eine 3D-Darstellung der Qualitätsfunktion gewählt.


```
In[14]:= Plot3D[f[x, y], {x, 0, 500}, {y, 0, 500},
  stelle Funktion graphisch in 3D dar
  ColorFunction → Hue, AxesLabel → {"x", "y", "f(x,y)"}]
  Farbfunktion Far... Achsenbeschriftungen
```

Out[14]=



Es lässt sich feststellen, dass die Funktion $f(x,y)$ genau ein Minimum aufweist (roter Bereich). Daraus kann man schließen, dass genau ein Paar von Koordinaten (x, y) existiert, das zur optimalen Lösung des Problems führt.

4.3 Die Nebenbedingungen

4.3.1 Die Nebenbedingung I

```
In[15]:= (*Nebenbedingung aus Aufgabenstellung*)
  (x - 100)^2 + (y - 100)^2 ≤ 400
```

Out[15]=

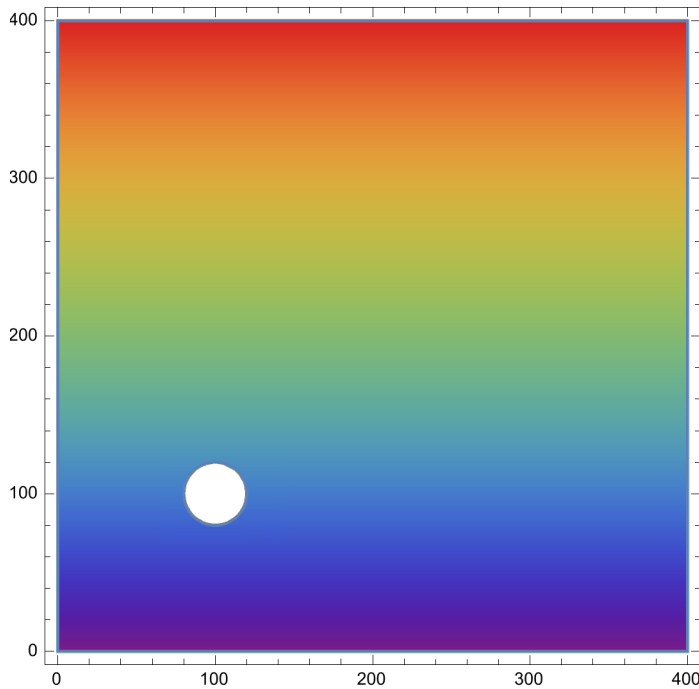
$$(-100 + x)^2 (-100 + y)^2 \leq 400$$

Die Nebenbedingung aus der Aufgabenstellung definiert die Umrandung und den Inhalt einer kreisförmigen Fläche (sogenannter See). Der zu bestimmende Flugplatz G darf nicht innerhalb des Sees bzw. am Rand des Sees liegen.

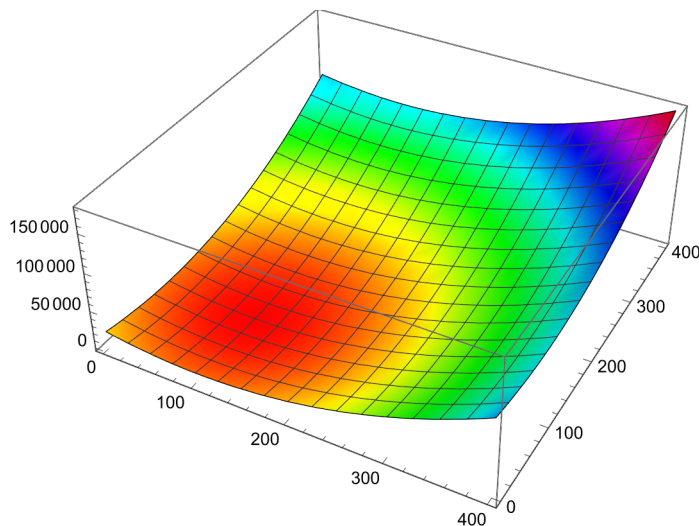
```
In[16]:= RegionPlot[(x - 100)^2 + (y - 100)^2 - 400 > 0,
  graphische Darstellung einer Region
  {x, 0, 400}, {y, 0, 400}, ColorFunction → "Rainbow"]
  Farbfunktion

Plot3D[(x - 100)^2 + (y - 100)^2 - 400, {x, 0, 400}, {y, 0, 400}, ColorFunction → Hue]
  stelle Funktion graphisch in 3D dar Farbfunktion Farbt...
```

Out[16]=



Out[17]=



Die Umformung der *Nebenbedingung aus der Aufgabenstellung* (neu benannt: *Nebenbedingung I*) dient dazu den Kreis mit Radius $r = 20$ auf den xy-Koordinaten $(100 | 100)$ als *feasible region* auszuschließen.

4.3.2 Die Nebenbedingung II

In[20]:= **(*Nebenbedingung aus Aufgabenstellung*)**

$$x + y \geq 250$$

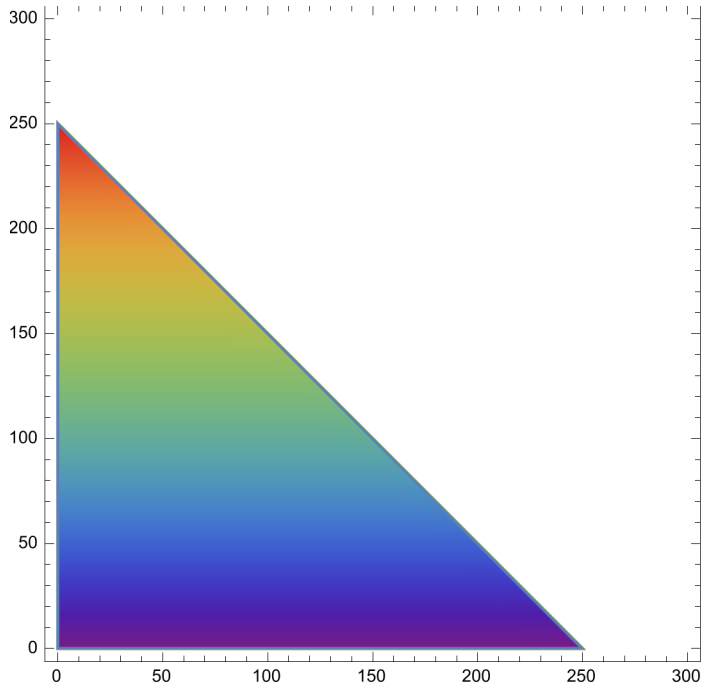
Out[20]=

$$x + y \geq 250$$

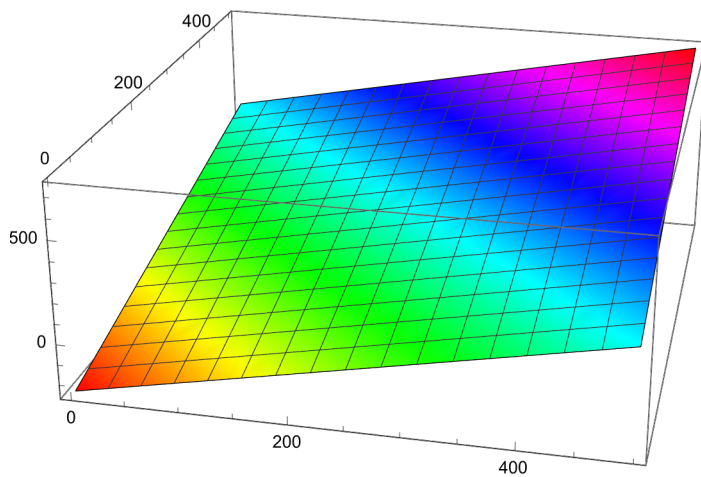
Die *Nebenbedingung aus der Aufgabenstellung* (neu benannt: *Nebenbedingung II*) definiert ein Sumpfgebiet (Gerade und Fläche). Der zu bestimmende Flugplatz G darf nicht innerhalb des Sumpfgebietes liegen .

```
In[21]:= RegionPlot[x + y - 250 < 0, {x, 0, 300}, {y, 0, 300}, ColorFunction -> "Rainbow"]
[graphische Darstellung einer Region] [Farbfunktion]
Plot3D[x + y - 250, {x, 0, 500}, {y, 0, 500}, ColorFunction -> Hue]
[stelle Funktion graphisch in 3D dar] [Farbfunktion] [Farbton]
```

Out[21]=



Out[22]=

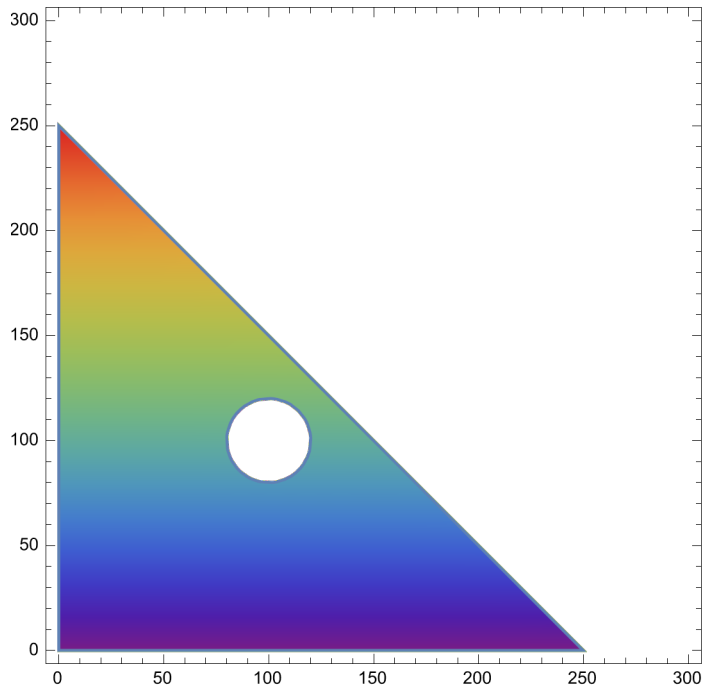


Die Umformung der *Nebenbedingung aus der Aufgabenstellung* (neu benannt: *Nebenbedingung II*) dient dazu die Fläche unterhalb der Geraden für einen neuen Flugplatz einzugrenzen.

4.3.3 Die Nebenbedingung I und Nebenbedingung II

```
In[23]:= RegionPlot[ $((x - 100)^2 + (y - 100)^2 - 400 > 0) \&\& (x + y - 250 < 0)$ ,  
graphische Darstellung einer Region  
{x, 0, 300}, {y, 0, 300}, ColorFunction -> "Rainbow"]  
Farbfunktion
```

Out[23]=



Die visualisierte Fläche im ersten Quadranten ist der eingegrenzte Standortraum, sog. *feasible region*.

4.4 Der optimale Flugplatz G

```
In[24]:= (* Constraint function 1 *)
h[x_, y_] := (x - 100)^2 + (y - 100)^2 - 400
h[x, y]

(* Constraint function 2 *)
g[x_, y_] := x + y - 250
g[x, y]
```

```
NMinimize[{f[x, y], g[x, y] < 0, h[x, y] > 0}, {x, y}, WorkingPrecision -> 10]
|minimiere numerisch |Arbeitspräzision
```

```
Out[25]=
-400 + (-100 + x)^2 + (-100 + y)^2
```

```
Out[27]=
-250 + x + y
```

```
Out[28]=
{13 956.22334, {x -> 123.3181792, y -> 126.6816467}}
```

Durch NMinimize kann man den optimalen Punkt für Flugplatz G unter den Nebenbedingungen I und II errechnen. Die Koordinaten (123.308 | 126.691) bezeichnen den mathematisch optimalen Punkt.

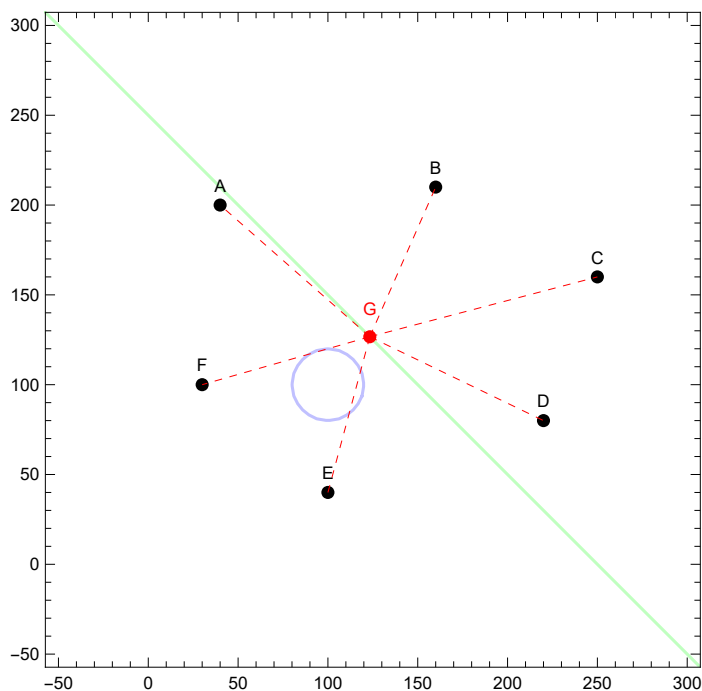
```
In[29]:= result = NMinimize[{f[x, y], g[x, y] < 0, h[x, y] > 0}, {x, y}, WorkingPrecision -> 10];
|minimiere numerisch |Arbeitspräzision
optimalX = x /. result[[2]];
optimalY = y /. result[[2]];
labels = {"A", "B", "C", "D", "E", "F"};
|Ko... |leit... |Exponentialkonstante E
```

```

In[33]:= (* Grafische Darstellung des optimalen Punktes
mit Nebenbedingungen I und II und den Flugplätzen A-F*)
[imaginäre Einheit I]
Show[{ContourPlot[x + y - 250 == 0, {x, -200, 500},
[zeige... [Konturgraphik]
{y, -200, 500}, ContourStyle -> {Opacity[0.25], Green}],
[Konturenstil [Deckkraft [grün]
ContourPlot[(x - 100)^2 + (y - 100)^2 - 400 == 0, {x, -200, 500},
[Konturgraphik]
{y, -200, 500}, ContourStyle -> {Opacity[0.25], Blue}],
[Konturenstil [Deckkraft [blau]
Graphics[{PointSize[0.02], Point[coordinates]}], Graphics[{PointSize[0.02], Red,
[Graphik [Punktgröße [Punkt [Graphik [Punktgröße [rot]
Point[{optimalX, optimalY}], Text["G", {optimalX, optimalY + 10}, {0, -1}]}],
[Punkt [Text]
Graphics[{Dashed, Red, Line[{optimalX, optimalY}, #]} & /@ coordinates]],
[Graphik [gestrichelt [rot [Linie]
Graphics[MapThread[Text[#1, #2, {0, -2}] &, {labels, coordinates}]]
[Graphik [wende auf ... [Text]
}], PlotRange -> {{-50, 300}, {-50, 300}}]
[Koordinatenbereich der Graphik]

```

Out[33]=



Der optimale Punkt G scheint auf der Geraden zu liegen.

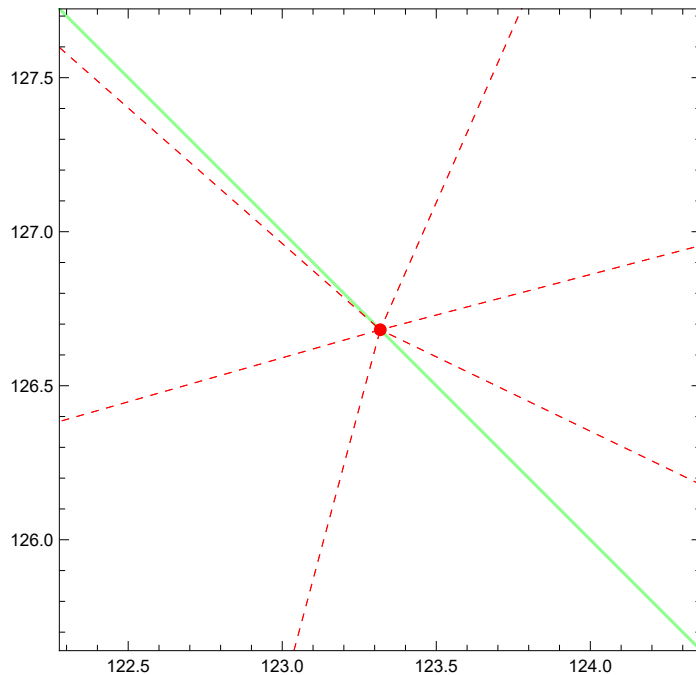
In[34]:=

```

(* Grafische Darstellung des optimalen Punktes mit
   Nebenbedingungen I und II und den Flugplätzen A-F mit PlotRange *)
[imaginäre Einheit] [Koordinatenbereich der Graphik]
Show[{ContourPlot[x + y - 250 == 0, {x, -200, 500}, {y, -200, 500},
[zeige...] [Konturgraphik]
      ContourStyle → {Opacity[0.25], Green}], ContourPlot[(x - 100)^2 + (y - 100)^2 - 400 == 0,
[Konturenstil] [Deckkraft] [grün] [Konturgraphik]
      {x, -200, 500}, {y, -200, 500}, ContourStyle → {Opacity[0.25], Blue}],
[Konturenstil] [Deckkraft] [blau]
      Graphics[{PointSize[0.02], Point[coordinates]}],
[Graphik] [Punktgröße] [Punkt]
      Graphics[{PointSize[0.02], Red, Point[{optimalX, optimalY}],
[Graphik] [Punktgröße] [rot] [Punkt]
      Text["Optimal G", {optimalX, optimalY + 10}, {0, -1}]}],
[Text]
      Graphics[{Dashed, Red, Line[{optimalX, optimalY}, #] & /@ coordinates}],
[Graphik] [gestrich...] [rot] [Linie]
      Graphics[MapThread[Text[#1, #2, {0, -2}] &, {labels, coordinates}]]
[Graphik] [wende auf ...] [Text]
    }, PlotRange → {{optimalX - 1, optimalX + 1}, {optimalY - 1, optimalY + 1}}]
[Koordinatenbereich der Graphik]

```

Out[34]=



```

In[35]:= NMinimize[{f[x, y], g[x, y] ≤ 0, h[x, y] ≥ 0}, {x, y}, WorkingPrecision → 10]
[minimiere numerisch] [Arbeitspräzision]

```

Out[35]=

```
{13 956.22334, {x → 123.3181792, y → 126.6816467}}
```

```
In[36]:= result2 = NMinimize[{f[x, y], g[x, y] ≤ 0, h[x, y] ≥ 0}, {x, y}, WorkingPrecision → 10];
           [minimiere numerisch] [Arbeitspräzision]
```

```
optimal2X = x /. result2[[2]];
optimal2Y = y /. result2[[2]];

```

```
(optimal2X + optimal2Y) < 250

```

```
optimal2X + optimal2Y

```

```
Out[39]=
```

```
True

```

```
Out[40]=
```

```
249.9998259

```

In den grafischen Darstellungen erweckt es den Anschein als ob der optimale Punkt G die Gerade der *Nebenbedingung aus der Aufgabenstellung* schneidet. Durch die obere Berechnung können wir jedoch feststellen, dass es keinen Schnittpunkt gibt. Dafür wurde die Funktion *WorkingPrecision* mit 10 Nachkomma-Stellen genutzt und die gefundenen Koordinaten in die Nebenbedingung eingesetzt.

5 Sensitivitätsanalyse

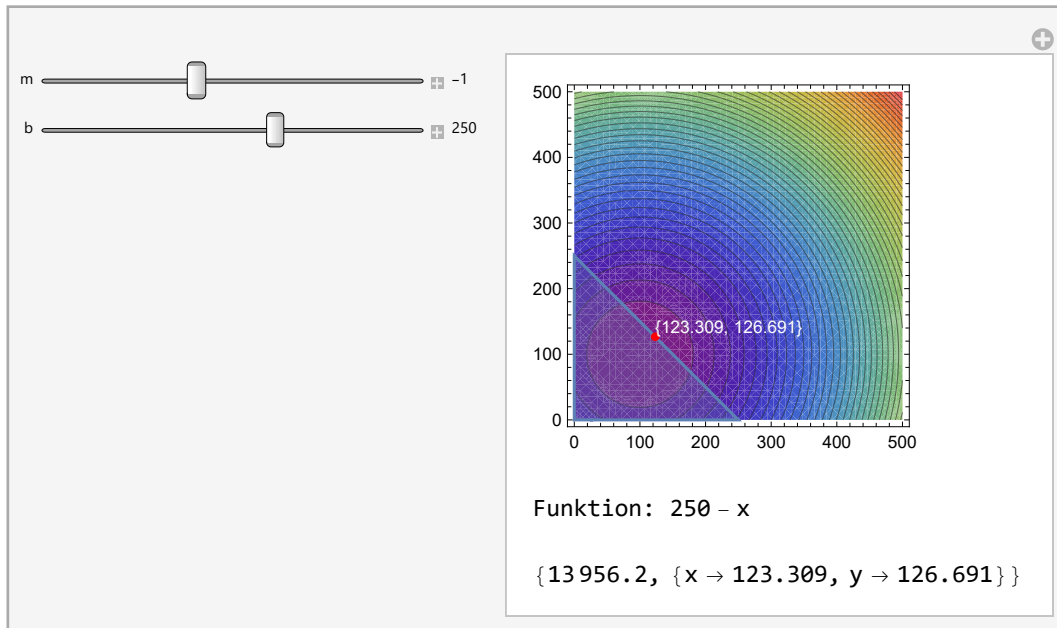
```
In[41]:= (* zi steht für die Anzahl der Flüge zu den Einzelnen Flughäfen;
coordinates beinhaltet die Koordinaten der Flughäfen A bis F;
labels steht für die Bezeichnungen der gegebenen Flugplätze *)
zi = {40, 10, 20, 30, 20, 10};
coordinates = {{40, 200}, {160, 210}, {250, 160}, {220, 80}, {100, 40}, {30, 100}};
labels = {"A", "B", "C", "D", "E", "F"};
      |Ko... |leit... |Exponentialkonstante E
```

```
(* Qualitätsfunktion zu euklidischen Abständen*)
f[x, y] := Total[zi * Sqrt[(coordinates[[All, 1]] - x]^2 + (coordinates[[All, 2]] - y)^2]]
      |Gesamts... |Quadratwurzel |alle |alle
f[x_, y_]
      |Ko... |leit... |Exponentialkonstante E
```

```
Out[45]=
20  $\sqrt{(100 - x_-)^2 + (40 - y_-)^2}$  + 30  $\sqrt{(220 - x_-)^2 + (80 - y_-)^2}$  + 10  $\sqrt{(30 - x_-)^2 + (100 - y_-)^2}$  +
20  $\sqrt{(250 - x_-)^2 + (160 - y_-)^2}$  + 40  $\sqrt{(40 - x_-)^2 + (200 - y_-)^2}$  + 10  $\sqrt{(160 - x_-)^2 + (210 - y_-)^2}$ 
```

```
In[46]:= Manipulate[
  |manipuliere
  Sol = NMinimize[{f[x, y], (x - 100)^2 + (y - 100)^2 - 400 > 0, y < m * x + b}, {x, y}];
      |minimiere numerisch
  Column[{Show[{ContourPlot[(x - 100)^2 + (y - 100)^2 - 400, {x, 0, 500}, {y, 0, 500},
    |Spalte |Zeige... |Konturgraphik
    ColorFunction -> "Rainbow", Contours -> 50, Epilog -> {Red, PointSize[0.025],
      |Farbfunktion |Konturen |Epilog |rot |Punktgröße
      Point[Sol[[2]][[All, 2]], White, Text[Sol[[2]][[All, 2]], Sol[[2]][[All, 2]], {-1, -1}]}],
      |Punkt |alle |weiß |Text |alle |alle
      , RegionPlot[y < m * x + b, {x, 0, 500}, {y, 0, 500}]}], ImageSize -> 200],
      |graphische Darstellung einer Region |Bildgröße
      "", Row[{"Funktion: ", m * x + b}], "", Sol}],
      |Zeile
  {{m, -1, "m"}, -5, 5, 0.1, Appearance -> "Labeled"},
      |Erscheinungsbild |beschriftet
  {{b, 250, "b"}, -1000, 1000, 1, Appearance -> "Labeled"},
      |Erscheinungsbild |beschriftet
  TrackedSymbols -> {m, b}, ControlPlacement -> Left
      |verfolgte Symbole |Platzierung des Bedien... |links
  ]
```

Out[46]=



Mithilfe der Funktion *Manipulate* können die Parameter der Nebenbedingungen I und II verändert werden. Die Auswirkungen auf den optimalen Flugplatz G sind umgehend sichtbar.

Wenn der Wert von “m” verändert wird, ändert sich die Steigung der Nebenbedingung II. Wenn der Wert von “b” verändert wird, verschiebt sich die Nebenbedingung entlang der y-Achse. Durch diese Änderungen können sich der zulässige Bereich und somit auch der optimale Punkt verschieben.

Wenn die Nebenbedingung nach oben verschoben wird kann sich der zulässige Bereich verengen und der optimale Punkt kann sich nach oben bewegen, um sich an den neuen zulässigen Bereich anzupassen. Wenn die Nebenbedingung hingegen nach unten verschoben wird, erweitert sich der zulässige Bereich und der optimale Punkt kann stehen bleiben, wenn er sich immer noch innerhalb des erweiterten zulässigen Bereichs befindet.

Das Optimum der Qualitätsfunktion ohne Nebenbedingung liegt im Punkt(132.555| 132.814).

6 Mögliche Verallgemeinerungen des Problems

Die oben beschriebene Herangehensweise kann für weitere Optimierungsprobleme in der Standortplanung übertragen werden.

6.1 Optimale Standorte zur Auslieferung von Paketen mit Flugdrohnen

Die Qualitätsfunktion für euklidische Abstände ist für Optimierungsprobleme nutzbar, welche Distanzen ohne Umwege (Luftlinie) zur Grundlage haben wie z.B. Flugdrohnenlieferungen im Luftraum.

Amazon und Google arbeiten seit 2012 an diesem Vorhaben und sind bereits in die Pilotphase zur Drohnenauslieferung von Paketen gestartet. Seit Dezember 2022 werden Lieferungen über Prime Air an 2 Standorten in den USA bereits den Kunden zum Test angeboten.

(Vgl. Donath, A. (2022): Amazon hat Drohnenlieferungen aufgenommen, online unter: <https://www.golem.de/news/prime-air-amazon-hat-drohnenlieferungen-aufgenommen-2212-170761.html>, [Zugriff am 14.06.2023])

Um zukünftig optimale Standorte zu finden bzw. die Infrastruktur für Auslieferungen aufzubauen, könnte die oben beschriebene Herangehensweise genutzt werden. Als einschränkende Nebenbedingungen wären Faktoren wie z. B. die Reichweite der Drohnen, die Nähe zum Kunden bzw. die anderweitige Nutzung des Flugraums denkbar.

6.2 Optimale Standorte für Elektro-Ladestationen der BVG

Im Rahmen der Umwelt- und Nachhaltigkeitsstrategie der Berliner Verkehrsbetriebe wird die Busflotte der BVG derzeit auf Elektro-Mobilität umgestellt. Dies erfordert zum Einen die Umrüstung der Fahrzeuge, zum Anderen den Aufbau einer Infrastruktur zum Laden bzw. zum Tausch der Fahrzeugbatterien.

(Vgl. Berliner Verkehrsbetriebe (BVG)(2023): Elektromobilität, online unter: <https://unternehmen.bvg.de/elektromobilitaet/> [Zugriff am 14.06.2023])

Zukünftig sollen weitere Betriebshöfe in der Innenstadt von Berlin aufgebaut werden. Zur Suche nach den optimalen Standorten kann die oben beschriebene Qualitätsfunktion um die Manhattan-Distanz (Taxi-Metrik) angepasst werden. Als einschränkende Nebenbedingungen wären hierbei die bestehende Bus- und Netzinfrastruktur, Reichweiten der Fahrzeugbatterien, Ruhezeiten von Fahrern sowie die Größe der zur Verfügung stehenden Immobilien im Stadtgebiet Berlin denkbar.

7 Schlussfolgerungen und Empfehlungen

Unsere Berechnungen des optimalen Flugplatzes G dienen als Grundlage für die Grobplanung eines neuen Flugplatzes. Der mathematisch aufgezeigte Punkt kann als *feasible region* verstanden werden. In einem Suchradius von etwa 10 km könnte der Aufbau eines neuen Flugplatzes in einer Machbarkeitsstudie näher untersucht werden.

Im weiteren Projektverlauf werden verschiedene Instrumente und Methoden (z.B. geografische, topologische und geologische Untersuchungen der *feasible region*) Aufschluss darüber geben, ob eine solch große Infrastruktur wie ein Flugplatz (Landebahn, Hangars, Air traffic control tower, Wendeschleifen, etc.) umsetzbar ist. Hinzu kommen rechtliche Rahmenbedingungen und sicherheitsrelevante Regularien (z.B. vorgegebene Entfernungen zu angrenzenden Gebäuden, Landesgrenzen oder Infrastrukturen), die berücksichtigt werden müssen.

Erst in der Feinplanung lässt sich der genaue Standort des neuen Flugplatzes bestimmen.

8 Zusammenfassung der Ergebnisse

Zusammenfassend lässt sich feststellen, dass die genutzte Qualitätsfunktion zu gewichteten euklidischen Abständen zu einem optimalen Flugplatz G unter Einhaltung aller Nebenbedingungen geführt hat. Das Ziel die zurückgelegten Flugkilometer zu minimieren ist gelungen. Der mathematisch aufgezeigte Punkt kann als *feasible region* verstanden werden. In einem Suchradius von etwa 10 km könnte der Aufbau eines neuen Flugplatzes in einer Machbarkeitsstudie näher untersucht werden.

Die Sensitivitätsanalyse hat gezeigt, dass die Methode robust ist. Daher kann geschlussfolgert werden, dass die Methode auch mit einer unendlichen Anzahl von weiteren gegebenen Flugplätzen funktioniert. Generell kann die Herangehensweise auf ähnliche Optimierungsprobleme mit euklidischen Abständen (z. B. Drohnenstandorte, Standorte für Ladeinfrastruktur) übertragen werden.

In der Realität ist die Standortplanung eines Flugplatzes ein komplexes Vorhaben und wird von einer Vielzahl an weiteren Nebenbedingungen (siehe *Schlussfolgerungen und Empfehlungen*) beeinflusst.

Generell leistet unsere Herangehensweise einen Beitrag zur Einsparung von fossilen Ressourcen und wirkt damit im Sinne der Nachhaltigkeit und Ressourcenschonung.