

# Wahlbezirkoptimierung mithilfe von Linear Programming

Marco Romanutti<sup>1,2</sup>

<sup>1</sup>Fachhochschule Nordwestschweiz FHNW, Brugg

<sup>2</sup>Effiziente Algorithmen, Klasse 5Id

In den USA wird bei Präsidentschaftswahlen mittels Electoral College gewählt. In fast allen 50 Bundesstaaten gilt das Prinzip „The winner takes it all“. Die Stimmenmehrheit in einem Wahlbezirk gewinnt für die entsprechende Partei einen Punkt, und die Partei mit mehr Punkten gewinnt die Präsidentschaftswahl.

Anhand von zwei Input-Dateien mit der Anzahl demokratischen, resp. republikanischen Stimmen für jedes Koordinatenfeld sollen 10 in etwa gleich grosse Wahlbezirke bestimmt werden. Abbildung 1 zeigt die effektiven Wahlbezirke nach der Wahl zum Repräsentantenhaus der Vereinigten Staaten im Jahr 2018.

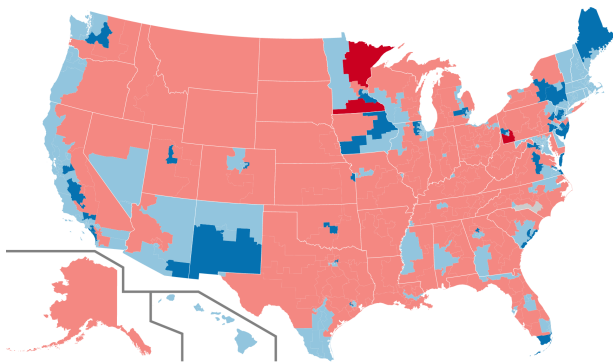


Figure 1: Wahlbezirke USA, Stand 2018

Die Wahlbezirke sollen so aufgeteilt werden, dass der Wahlsieg möglichst deutlich ausfällt für

- die republikanischen Kandidaten
- die demokratischen Kandidaten

## 1 Modell

Das Modell für einen möglichst deutlichen Wahlsieg in  $B$  Wahlbezirken ist in den nachfolgenden Abschnitten beschrieben.

### Vars

$VD_b$  = Wahlsieg für Demokraten in Bezirk  $b$

$VR_b$  = Wahlsieg für Republikaner in Bezirk  $b$

$C_{xyb}$  = Bezirk  $b$  von Koordinate  $(x,y)$

$R_b$  = Stimmen für Republikaner in Bezirk  $b$

$D_b$  = Stimmen für Demokraten in Bezirk  $b$

Wobei gilt:

$$VD_b = \begin{cases} 1, & D_b > R_b \\ 0, & \text{sonst} \end{cases} \quad (1)$$

$$VR_b = \begin{cases} 1, & R_b > D_b \\ 0, & \text{sonst} \end{cases} \quad (2)$$

Das Setzen des Werts 0 oder 1 kann mit Ungleichung (4) erreicht werden (am Beispiel eines Wahlsiegs für die Demokraten):

$$VD_b = D_b \stackrel{?}{>} R_b \quad (3)$$

$$0 \leq R_b - D_b + M * VD_b \leq M - 1 \quad (4)$$

Der Wert für  $M$  wurde anhand der grössten Differenz  $R_b - D_b$  von allen Bezirken  $b$  festgelegt<sup>1</sup>: Falls  $D_b$  grösser ist als  $R_b$  ist  $R_b - D_b$  negativ und  $VD_b$  muss den Wert 1 haben. Falls  $D_b$  hingegen nicht grösser ist als  $R_b - D_b$  ergibt  $R_b - D_b$  einen nicht negativen Wert und  $VD_b$  muss 0 sein.

<sup>1</sup> $M$  = grösste Differenz  $R_b - D_b + 1$ , vgl. [2]

Aus Zeitgründen wurde dieser Wert als Konstante für alle  $b$  definiert. In einer optimierten Variante könnte dieser Wert individuell gesetzt werden.

Im Falle von gleich vielen Stimmen in einem Bezirk  $b$  erhält keine der Parteien eine Stimme. Das Total von Wahlsiegen in den verschiedenen Bezirken kann deshalb auch kleiner als  $B$  sein.

Die Bezirke  $b$  sind als Binärvariablen modelliert: Pro Koordinate  $(x,y)$  nimmt die Variable nur dann den Wert 1 an, falls der Index  $b$  dem zugewiesenen Bezirk entspricht.

$$C_{xyb} = \begin{cases} 1, & b = \text{zugew. Bezirk} \\ 0, & \text{sonst} \end{cases} \quad (5)$$

Die Stimmen von Koordinate  $(x,y)$  für Republikaner<sup>2</sup>, resp. jene für die Demokraten<sup>3</sup> wurden nicht als Variablen im Modell definiert.  $R_b$  und  $D_b$  sind im Gurobi-Modell als lineare Ausdrücke implementiert<sup>4</sup>.

## Constraints

(A) Jede Zelle mit den Koordinaten  $(x,y)$  gehört nur zu einem Bezirk

$$\forall x, y : \sum_b C_{xyb} = 1 \quad (6)$$

(B) Jeder Wahlbezirk muss mindestens 135 Zellen beinhalten

$$\forall b : \sum_{x,y} C_{xyb} \geq 135 \quad (7)$$

Mit dem Constraint (B) konnten die Wahlbezirke in etwa gleich gross gehalten werden.

Als zusätzliche Einschränkung wurden bereits nur Koordinaten  $(x,y)$  mit mehr als einer Stimme<sup>5</sup> für das Modell berücksichtigt. Diese Einschränkung ist jedoch nicht als Constraint im Modell definiert.

## Objective

Je nachdem, ob Teilaufgabe a) oder b) gelöst werden soll, muss nun eine andere Variable maximiert werden. Folgendes Beispiel zeigt den Fall von möglichst vielen Wahlsiegen für die Demokraten:

$$\max \sum_b V D_b \quad (8)$$

<sup>2</sup> $r_{xy}$  = Stimmen für Republikaner mit Koordinate  $(x,y)$

<sup>3</sup> $d_{xy}$  = Stimmen für Demokraten mit Koordinate  $(x,y)$

<sup>4</sup>z.B.  $D_b = \forall b : \sum_{x,y} C_{xyb} * d_{xy}$

<sup>5</sup> $\forall x, y : r_{xy} + d_{xy} \geq 1$

## 2 Anleitung Software

Die Modelle können mit der main-Methode der Klasse Election generiert werden. Die Outputs werden im mps-Format in das root-Verzeichnis geschrieben. Beim Aufruf der Methode winCount innerhalb der main-Methode kann angegeben werden, welche der Variablen  $V D_b$  und  $V R_b$  zum Modell hinzugefügt werden sollen. Im Gegensatz zum den Unit-Tests (vgl. Kapitel 3) wird aus Performancegründen nur die zu optimierende Variable zum Modell hinzugefügt<sup>6</sup>.

Mit den generierten Modellen konnte bei der Maximierung von demokratischen Wahlsiegen ein Gesamtscore von 10:0 aus Sicht der Demokraten erreicht werden. Bei der Maximierung von republikanischen Wahlsiegen wurde ein Gesamtscore von 7:3 aus Sicht der Republikaner erreicht.

## 3 Korrektheit

Die Korrektheit wurde mit verschiedenen Unit-Tests geprüft. Die dazugehörigen Input-Files im Verzeichnis src/test/java/resources enthalten die Anzahl Zeilen und Spalten, gefolgt von der jeweiligen Anzahl Stimmen für Demokraten oder Republikaner. Daraus ergibt sich folgendes Format:

```
10 1
0 0 0 0 0 1 1 1 1 1
```

Die nachfolgende Tabelle zeigt die durchgeführten Tests und erwarteten Resultate. Die Anzahl Siege der Demokraten werden mit  $D$  abgekürzt, jene der Republikaner mit  $R$ .

Table 1: Edge-Cases Election

Beschreibung	Resultat
[1] Input-File mit Anzahl Bezirke $< B$	Infeasible
[2] Gleiche Anzahl Bezirke mit derselben Anzahl Stimmen pro Bezirk (Unentschieden)	D 0, R 0
[3] Ein Bezirk mit gleicher Stimmenanzahl bei restlichen Siegen für Demokraten	D 9, R 0
[4] Alle Bezirke von Demokraten gewonnen, bei Maximierung von demokratischen Siegen	D 10, R 0
[5] Alle Bezirke von Demokraten gewonnen, bei Maximierung von republikanischen Siegen	D 10, R 0
[6] Alle Bezirke von Republikaner gewonnen, bei Maximierung von demokratischen Siegen	D 0, R 10
[7] Alle Bezirke von Republikaner gewonnen, bei Maximierung von republikanischen Siegen	D 0, R 10
[8] Je 5 Bezirke gewonnen, bei Maximierung von demokratischen Siegen	D 5, R 5
[9] Je 5 Bezirke gewonnen, bei Maximierung von republikanischen Siegen	D 5, R 5
[10] Zu kleines $M$ (vgl. Kapitel 1)	Infeasible

<sup>6</sup>Bei den Unit-Tests werden zur Überprüfung der Korrektheit beide Variablen hinzugefügt und ausgewertet

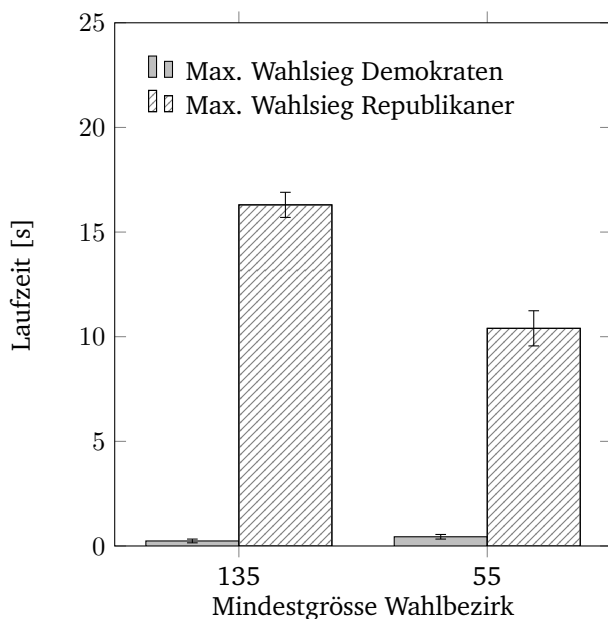
Zusätzlich zu den Edge-Cases werden die Resultate nach der Optimierung des Modells mittels `assert`-Statements überprüft. Dabei werden folgende Überprüfungen gemacht:

- **Überprüfen der Constraints:** Jede Zelle darf nur zu einem Bezirk gehören und jeder Bezirk muss mindestens 135 Zellen umfassen
- **Überprüfen von  $VD$  und  $RD$ :** Entsprechen die gesetzten Wahlsiege im Bezirk  $b$  den tatsächlichen Wahlresultaten in jenem Bezirk
- **Überprüfen der Anzahl Gewinner:** Total darf es nicht mehr als  $B$  Sieger von Wahlbezirken geben

## 4 Effizienz

Um die Laufzeit der zur Verfügung gestellten Input-Files zu messen, wurde das Modell jeweils 10-mal mit unterschiedlichen Seeds ausgeführt. Die durchschnittliche Laufzeit bei Optimierung nach demokratischen Wahlsiegen beträgt 0.24 s, jene bei Optimierung nach republikanischen Wahlsiegen 16.3 s. Die nachfolgende Grafik zeigt, dass mit unausgeglichene Wahlbezirken schnellere Laufzeiten erzielt werden konnten.

Figure 2: Laufzeiten der Modelle



## References

- [1] Wikipedia: Wahlen USA, [https://de.wikipedia.org/wiki/Wahl\\_zum\\_Repräsentantenhaus\\_der\\_Vereinigten\\_Staaten\\_2018](https://de.wikipedia.org/wiki/Wahl_zum_Repräsentantenhaus_der_Vereinigten_Staaten_2018)
- [2] ILP: Greater than comparison, <https://blog.adamfurmanek.pl/2015/09/12/ilp-part-4/>