Lösungsvorschläge

Präferenzaggregation durch Wählen: Algorithmik und Komplexität

Bearbeitungszeit: 09. November bis 16. November, 12:00 mittags
Verantwortlich: Christian Laußmann

Aufgabe 1 (10 Punkte): Komplexität Gewinnerbestimmung

Nehmen Sie an, dass die folgenden beiden Funktionen in konstanter Zeit (also $\mathcal{O}(1)$) berechnet werden können:

- $\mathrm{pos}_v(x)$ gibt die Position von Kandidat x in der Präferenz v zurück.
- $\mathrm{prefers}_v(x, y)$ gibt "True" zurück, wenn $x >_v y$ und "False" andernfalls.

(a) Geben Sie für *Copeland* und *Scoring-Protokolle* mit beliebigem Scoring-Vektor α je einen möglichst effizienten Algorithmus (Pseudocode) an, der als Eingabe die Menge der Präferenzen V und die Menge der Kandidaten C bekommt und unter Verwendung der Funktionen pos und prefers alle Gewinner der Wahl bestimmt.

(b) Geben Sie die (Worst-Case-) Laufzeit Ihrer Algorithmen in Abhängigkeit von $|V| = n$ und $|C| = m$ in Landau-Notation an. Begründen Sie.

(c) Was können Sie über die Komplexität der Manipulationsprobleme für Copeland und Scoring-Protokolle folgern?

Lösungsvorschlag:

(a) **Copeland:**

- Setze $s(c) := 0$ für alle $c \in C$.
- Für jeden Kandidaten c_i mit $i \in \{1, ..., |C| - 1\}$:

  | Für jeden Kandidaten c_j mit $j \in \{i + 1, ..., |C|\}$:

    | Setze $x := 0$.

    | Für jede Präferenz $v \in V$:

      | Wenn $\mathrm{prefers}_v(c_i, c_j)$, dann setze $x := x + 1$.

      | Sonst setze $x := x - 1$.

    | Wenn $x = 0$, dann setze $s(c_i) := s(c_i)+0.5$ und $s(c_j) := s(c_j)+0.5$.

    | Wenn $x > 0$, dann setze $s(c_i) := s(c_i) + 1$.

| Wenn $x < 0$, dann setze $s(c_j) := s(c_j) + 1$

- Gib alle Kandidaten c mit dem größten $s(c)$ als Gewinner zurück.

Scoring-Protokolle mit $\alpha = (\alpha_1, \alpha_2, ..., \alpha_m)$:

- Setze $s(c) := 0$ für alle $c \in C$.
- Für jeden Kandidaten $c \in C$:

  | Für jede Präferenz $v \in V$:

  | Setze $s(c) := s(c) + \alpha_{\text{pos}_v(c)}$.
- Gib alle Kandidaten c mit dem größten $s(c)$ als Gewinner zurück.

(b) In beiden Algorithmen initialisieren wir alle Scores mit 0 und wählen am Ende die Kandidaten mit höchsten Scores aus. Beides ist in $\mathcal{O}(m)$ möglich.

Der Copeland-Algorithmus hat drei verschachtelte Schleifen. Die Anweisungen in der innersten Schleife werden $m \cdot \frac{m-1}{2} \cdot n$ mal ausgeführt. Daraus ergibt sich die Gesamtlaufzeit von $\mathcal{O}(n \cdot m^2)$.

Der Scoring-Protokoll-Algorithmus hat zwei verschachtelte Schleifen. Eine in Abhängigkeit von m und eine in Abhängigkeit von n. Daraus ergibt sich die Laufzeit von $\mathcal{O}(n \cdot m)$.

(c) Die Gewinnerbestimmung geht bei Copeland und Scoring-Protokollen in Polynomialzeit. Nach Aufgabe 4.2 sind die Manipulationsprobleme für diese Wahlsysteme also in NP.

Aufgabe 2 (10 Punkte): Pareto Kriterium

Wir sagen Kandidat a *Pareto-dominiert* Kandidat b, wenn alle Wähler a mindestens so gut finden wie b und mindestens ein Wähler a echt besser findet als b. Wir behandeln im folgenden nur Präferenzen ohne Gleichstände. Für Präferenzen ohne Gleichstände ist Pareto-Dominanz äquivalent dazu, dass alle Wähler a vor b bevorzugen. Ein Wahlsystem erfüllt das *Pareto-Kriterium*, wenn ein Gewinner nie Pareto-dominiert wird.

(a) Zeigen Sie, dass Dodgson das Pareto-Kriterium erfüllt.

(b) Geben Sie ein Wahlsystem an, das das Pareto-Kriterium verletzt. Demonstrieren Sie dies an einem Beispiel.

Lösungsvorschlag:

(a) Sei $w \in C$ ein Dodgson-Gewinner, also ein Kandidat der mit der kleinsten Anzahl σ an Vertauschungen zum Condorcet-Gewinner gemacht werden kann.

Angenommen es gäbe nun einen Kandidaten $u \in C$, der w Pareto-dominiert, also in jeder Präferenz vor w steht. Wir geben eine Konstruktion mit weniger als σ Vertauschungen an, die u zum Condorcet-Gewinner macht, was ein Widerspruch zur Annahme ist, dass w mit der kleinsten Anzahl an Vertauschungen zum Condorcet-Gewinner gemacht werden kann.

Sei X eine kürzeste Folge an Vertauschungen, mit der wir w zum Condorcet-Gewinner machen können. Wir konstruieren die Folge Y, mit der u zum Condorcet-Gewinner wird, wie folgt:

- Bei allen Präferenzen, in denen wir durch X keine Änderungen vornehmen, nehmen wir auch mit Y keine Änderungen vor.
- Alle Vertauschungen aus X, die wir in einer Präferenz durchführen, übernehmen wir in Y bis zu dem Punkt, wo u und w getauscht werden sollen (da u im paarweisen Vergleich w schlägt, kommt dieser Fall mindestens $\frac{|V|}{2} + 1$ mal vor). Diesen Tausch lassen wir aus und fahren mit weiteren Vertauschungen in dieser Präferenz fort, aber anstatt w nach oben zu tauschen, tauschen wir ab jetzt u nach oben.

Die Folge Y hat echt weniger Vertauschungen als X, da wir die Vertauschungen von u und w auslassen. Weiterhin ist u durch diese Vertauschungen der Condorcet-Gewinner, denn in allen Präferenzen steht u nach den Vertauschungen in Y mindestens so gut da, wie w nach den Vertauschungen in X.

(b) 2-Approval verletzt das Pareto-Kriterium. Wenn beispielsweise alle Wähler die Präferenz $a > b > c > d$ haben, dann gewinnen a und b in 2-Approval. Kandidat a wird aber von allen Wählern gegenüber b bevorzugt, also wird der Gewinner b von a Pareto-dominiert.

Aufgabe 3 (12 Punkte): Maximin Wahlsystem

Aus der Vorlesung kennen Sie das Maximin Wahlsystem.

(a) Beweisen Sie, dass das Maximin Wahlsystem das Condorcet Kriterium erfüllt.

(b) Zeigen Sie mit einem Beispiel, dass im Maximin Wahlsystem auch ein Condorcet-Verlierer alleiniger Gewinner werden kann.

Lösungsvorschlag:

(a) Wenn ein Condorcet-Gewinner existiert, dann gewinnt er jeden paarweisen Vergleich. Bei n Wählern ist also der MScore eines Condorcet-Gewinners mindestens $\lfloor \frac{n}{2} \rfloor + 1$. Da jeder andere Kandidat zumindest den paarweisen Vergleich mit dem Condorcet-Gewinner verliert, kann jeder andere Kandidat höchstens einen MScore von $\lceil \frac{n}{2} \rceil - 1$ haben.

Damit gewinnt der Condorcet-Gewinner auch immer in Maximin Wahlen.

(b) Wir betrachten die folgenden Präferenzen:

$$\begin{gathered} l > a > b > c > d > e \\ l > e > a > b > c > d \\ d > e > a > b > c > l \\ c > d > e > a > b > l \\ b > c > d > e > a > l \end{gathered}$$

Die Kandidaten a, b, c, d, e bilden einen Zyklus, bei dem jeder Kandidat den nächsten jeweils mit $4:1$ im paarweisen Vergleich schlägt. Daraus folgt, dass a, b, c, d, e jeweils maximal einen MScore von 1 haben können.

Kandidat l verliert gegen jeden Kandidaten mit $2:3$. Er ist somit Condorcet-Verlierer, aber verliert jeweils nur knapp. Mit einem MScore von 2 gewinnt l die Wahl.

Aufgabe 4 (8 Punkte): Maximin Manipulation

Gegeben sei folgende Instanz für Maximin-CCWM:

$$\begin{aligned} C &= \{a, b, c\} \\ V &= (a >_1 b >_1 c \\ &\quad\;\; b >_2 a >_2 c \\ &\quad\;\; c >_3 b >_3 a) \end{aligned}$$

Die manipulierenden Wähler sind $\{4, 5, 6\}$ und die Gewichte sind $w(1) = 3, w(2) = 2, w(3) = 7, w(4) = 1, w(5) = 1, w(6) = 1$. Die Manipulatoren wollen Kandidat a zum *alleinigen* Sieger der Wahl machen.

(a) Geben Sie die Maximin-Scores aller Kandidaten vor der Manipulation an.

(b) Ist die Manipulation möglich? Falls ja, geben Sie Stimmen für Wähler $4, 5$ und 6 an, welche die Manipulation erfolgreich machen. Geben Sie außerdem die Maximin-Scores nach der Manipulation an. Falls nein, beweisen Sie, dass eine Manipulation nicht möglich ist.

Lösungsvorschlag:

(a) Die paarweisen Vergleiche vor der Manipulation gehen wie folgt aus:

$$\begin{gathered} N(a, b) = 3,\ N(a, c) = 5, \\ N(b, a) = 9,\ N(b, c) = 5, \\ N(c, a) = 7,\ N(c, b) = 7. \end{gathered}$$

Damit ergeben sich $MScore(a) = 3, MScore(b) = 5$ und $MScore(c) = 7$.

(b) Alle Manipulatoren haben a als oberstes in ihrer Präferenz (falls eine Manipulation erfolgreich ist, wo a nicht an erster Stelle steht, so machen wir a aber auch nicht schlechter wenn wir a nach oben tauschen).

Nach Vorlesung gilt: Wenn es eine Manipulation gibt, bei der jeder Manipulator entweder $a > b > c$ oder $a > c > b$ stimmt und a gewinnt, dann gibt es auch eine Manipulation bei der alle Manipulatoren gleich abstimmen und a gewinnt (Folie 2.25). Wir müssen also nur prüfen, ob die Manipulation erfolgreich ist, wenn alle Manipulatoren $a > b > c$ oder alle Manipulatoren $a > c > b$ stimmen.

Fall 1: Alle stimmen $a > b > c$.
Dann ändert sich $N(a, b) = 6, N(a, c) = 8$ und $N(b, c) = 8$. Damit folgt $MScore(a) = 6, MScore(b) = 8$ und $MScore(c) = 7$. Also gewinnt a bei dieser Manipulation nicht.

Fall 2: Alle stimmen $a > c > b$.
Dann ändert sich $N(a, b) = 6, N(a, c) = 8$ und $N(c, b) = 10$. Es folgt $MScore(a) = 6, MScore(b) = 5$ und $MScore(c) = 7$. Also gewinnt a auch bei dieser Manipulation nicht.

Also kann es keine Manipulation geben, bei der a gewinnt.