technische universität dortmund

Effiziente Algorithmen und Komplexitätstheorie /

Informatik

Aufteilungs- und Zuweisungsalgorithmen

Proseminar
im Wintersemester 2016/2017

Anja Rey, 19. Oktober 2016

Organisatorisches

Proseminar (2 CP)

Organisatorisches

Proseminar (2 CP)

- schriftliche Ausarbeitung zu einem ausgewählten Thema
  - 5 bis 7 Seiten, PDF-Format (LaTeX)
  - in eigenen Worten (reine Übersetzung nicht ausreichend)
  - formal und anschaulich
  - vollständige Quellenangaben

Organisatorisches

Proseminar (2 CP)

- schriftliche Ausarbeitung zu einem ausgewählten Thema
  - 5 bis 7 Seiten, PDF-Format (LaTeX)
  - in eigenen Worten (reine Übersetzung nicht ausreichend)
  - formal und anschaulich
  - vollständige Quellenangaben
- Rückmeldung zu einer anderen Ausarbeitung in Form eines *Peer-Reviews*

Organisatorisches

Proseminar (2 CP)

- schriftliche Ausarbeitung zu einem ausgewählten Thema
  - 5 bis 7 Seiten, PDF-Format (LaTeX)
  - in eigenen Worten (reine Übersetzung nicht ausreichend)
  - formal und anschaulich
  - vollständige Quellenangaben
- Rückmeldung zu einer anderen Ausarbeitung in Form eines *Peer-Reviews*
- erfolgreich absolviertes Zwischengespräch
  - Rückmeldung der Seminarleiterin
  - Vorstellung eines schlüssigen Vortragskonzepts

Organisatorisches

Proseminar (2 CP)

- schriftliche Ausarbeitung zu einem ausgewählten Thema
  - 5 bis 7 Seiten, PDF-Format (LaTeX)
  - in eigenen Worten (reine Übersetzung nicht ausreichend)
  - formal und anschaulich
  - vollständige Quellenangaben
- Rückmeldung zu einer anderen Ausarbeitung in Form eines *Peer-Reviews*
- erfolgreich absolviertes Zwischengespräch
  - Rückmeldung der Seminarleiterin
  - Vorstellung eines schlüssigen Vortragskonzepts
- 30-minütiger Vortrag des Themas (+10 min für Fragen)

Organisatorisches

Proseminar (2 CP)

- schriftliche Ausarbeitung zu einem ausgewählten Thema
  - 5 bis 7 Seiten, PDF-Format (LaTeX)
  - in eigenen Worten (reine Übersetzung nicht ausreichend)
  - formal und anschaulich
  - vollständige Quellenangaben
- Rückmeldung zu einer anderen Ausarbeitung in Form eines *Peer-Reviews*
- erfolgreich absolviertes Zwischengespräch
  - Rückmeldung der Seminarleiterin
  - Vorstellung eines schlüssigen Vortragskonzepts
- 30-minütiger Vortrag des Themas (+10 min für Fragen)
- Frage zu mindestens einem Vortrag

Organisatorisches

Proseminar (2 CP)

- schriftliche Ausarbeitung zu einem ausgewählten Thema
  - 5 bis 7 Seiten, PDF-Format (LaTeX)
  - in eigenen Worten (reine Übersetzung nicht ausreichend)
  - formal und anschaulich
  - vollständige Quellenangaben
- Rückmeldung zu einer anderen Ausarbeitung in Form eines *Peer-Reviews*
- erfolgreich absolviertes Zwischengespräch
  - Rückmeldung der Seminarleiterin
  - Vorstellung eines schlüssigen Vortragskonzepts
- 30-minütiger Vortrag des Themas (+10 min für Fragen)
- Frage zu mindestens einem Vortrag

Präsentationskurs (1 CP)

begleitend: Do, 14–16 Uhr in OH 14, E 02 vom 20. Oktober bis zum 8. Dezember

Zeitplan

Beginn

- Mi 19.10.16 Einleitung, Themen
- Do 20.10.16 Beginn Präsentationskurs

Zeitplan

Beginn

- Mi 19.10.16 Einleitung, Themen
- Do 20.10.16 Beginn Präsentationskurs

Themenvergabe

- Thema auswählen bis 25.10 unter
  `http://doodle.com/poll/37hfa3gxyn4xdc7n`

Zeitplan

Beginn

- Mi 19.10.16 Einleitung, Themen
- Do 20.10.16 Beginn Präsentationskurs

Themenvergabe

- Thema auswählen bis 25.10 unter

  `http://doodle.com/poll/37hfa3gxyn4xdc7n`

Ausarbeitungsphase

- Mi 26.10.16 Besprechung
- Mi 02.11.16 & Mi 09.11.16 Fragen & Bearbeitung
- Di 15.11.16 erste Abgabe Ausarbeitung

— Rechtzeitig Vortragsplanung beginnen! —

Zeitplan

Beginn

- Mi 19.10.16 Einleitung, Themen
- Do 20.10.16 Beginn Präsentationskurs

Themenvergabe

- Thema auswählen bis 25.10 unter

  `http://doodle.com/poll/37hfa3gxyn4xdc7n`

Ausarbeitungsphase

- Mi 26.10.16 Besprechung
- Mi 02.11.16 & Mi 09.11.16 Fragen & Bearbeitung
- Di 15.11.16 erste Abgabe Ausarbeitung

— Rechtzeitig Vortragsplanung beginnen! —

Reviewphase

- Verteilung Reviews (per E-Mail)
- Mi 16.11.16 Besprechung
- Mi 23.11.16 Fragen & Bearbeitung
- Do 24.11.16 Abgabe Reviews
- Mi 30.11.16 Fragen & Bearbeitung
- Mi 30.11.16 zweite Abgabe Ausarbeitung

Zeitplan

Beginn

- Mi 19.10.16 Einleitung, Themen
- Do 20.10.16 Beginn Präsentationskurs

Themenvergabe

- Thema auswählen bis 25.10 unter
  `http://doodle.com/poll/37hfa3gxyn4xdc7n`

Ausarbeitungsphase

- Mi 26.10.16 Besprechung
- Mi 02.11.16 & Mi 09.11.16 Fragen & Bearbeitung
- Di 15.11.16 erste Abgabe Ausarbeitung

— Rechtzeitig Vortragsplanung beginnen! —

Reviewphase

- Verteilung Reviews (per E-Mail)
- Mi 16.11.16 Besprechung
- Mi 23.11.16 Fragen & Bearbeitung
- Do 24.11.16 Abgabe Reviews
- Mi 30.11.16 Fragen & Bearbeitung
- Mi 30.11.16 zweite Abgabe Ausarbeitung

Zwischenbesprechung

- 8. Woche Einzeltermine
- Do 08.12.16 Ende Präsentationskurs

Zeitplan

Beginn

- Mi 19.10.16 Einleitung, Themen
- Do 20.10.16 Beginn Präsentationskurs

Themenvergabe

- Thema auswählen bis 25.10 unter
  `http://doodle.com/poll/37hfa3gxyn4xdc7n`

Ausarbeitungsphase

- Mi 26.10.16 Besprechung
- Mi 02.11.16 & Mi 09.11.16 Fragen & Bearbeitung
- Di 15.11.16 erste Abgabe Ausarbeitung

— Rechtzeitig Vortragsplanung beginnen! —

Reviewphase

- Verteilung Reviews (per E-Mail)
- Mi 16.11.16 Besprechung
- Mi 23.11.16 Fragen & Bearbeitung
- Do 24.11.16 Abgabe Reviews
- Mi 30.11.16 Fragen & Bearbeitung
- Mi 30.11.16 zweite Abgabe Ausarbeitung

Zwischenbesprechung

- 8. Woche Einzeltermine
- Do 08.12.16 Ende Präsentationskurs

Vortragsphase und Ende

- 9.–15. Woche: Vorträge
- Feedback am Ende

Fragen?

- Informationen unter

  `http://ls2-www.cs.uni-dortmund.de/~rey/wise1617proseminar`

- Abgaben und weitere Fragen an anja.rey@tu-dortmund.de
- Zwischengespräch und weitere Fragen in OH 14, 313
- weitere Fragen in Seminarterminen und Sprechstunden

  Mo 13:30 Uhr bis 14:30 Uhr

Themenübersicht

Aufteilungsalgorithmen

- faire Aufteilung teilbarer Güter
- faire Aufteilung unteilbarer Güter

Zuweisungsalgorithmen

- das HOSPITALS-RESIDENTS-Problem
- das Mitbewohner-Problem
- Häuser zuordnen
- hedonische Spiele

Aufteilung und Zuweisung mit Präferenzen

Eine Instanz besteht aus

- Agenten (häufig zwei disjunkte Mengen)
- ordinalen Präferenzen über Teilmengen der Agenten
- anderen Bedingungen: Kapazitäten etc.

Aufteilung und Zuweisung mit Präferenzen

Eine Instanz besteht aus

- Agenten (häufig zwei disjunkte Mengen)
- ordinalen Präferenzen über Teilmengen der Agenten
- anderen Bedingungen: Kapazitäten etc.

Ziel: faire oder optimale Aufteilung (Allocation) oder Zuweisung (Matching) finden

Aufteilung und Zuweisung mit Präferenzen

Eine Instanz besteht aus

- Agenten (häufig zwei disjunkte Mengen)
- ordinalen Präferenzen über Teilmengen der Agenten
- anderen Bedingungen: Kapazitäten etc.

Ziel: faire oder optimale Aufteilung (Allocation) oder Zuweisung (Matching) finden

Wir unterscheiden zwischen

- bipartiten Zuweisungen mit beidseitigen Präferenzen
- bipartiten Aufteilungen mit einseitigen Präferenzen
  - Güter: teilbar (sharable, divisible) vs. nicht teilbar
- nicht-bipartiten Zuweisungen mit Präferenzen

Aufteilung und Zuweisung mit Präferenzen

Eine Instanz besteht aus

- Agenten (häufig zwei disjunkte Mengen)
- ordinalen Präferenzen über Teilmengen der Agenten
- anderen Bedingungen: Kapazitäten etc.

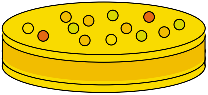
Ziel: faire oder optimale Aufteilung (Allocation) oder Zuweisung (Matching) finden

Wir unterscheiden zwischen

- bipartiten Zuweisungen mit beidseitigen Präferenzen
- bipartiten Aufteilungen mit einseitigen Präferenzen
  - Güter: teilbar (sharable, divisible) vs. nicht teilbar
- nicht-bipartiten Zuweisungen mit Präferenzen

Hier: Design und Analyse effizienter Algorithmen (oder ggf. Aussagen über Nicht-Existenz oder Härte solcher Algorithmen)

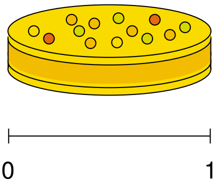
Aufteilung teilbarer Güter



Situation: inhomogener Kuchen

Ziel: jeder bekommt eine faire Portion

Aufteilung teilbarer Güter



Situation: inhomogener Kuchen

Ziel: jeder bekommt eine faire Portion

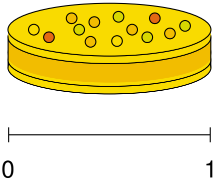
Definition 1

Eine Cake-Cutting-Instanz besteht aus einem Kuchen $[0, 1]$,

Aufteilung teilbarer Güter

Situation: inhomogener Kuchen

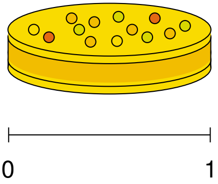
Ziel: jeder bekommt eine faire Portion



Definition 1

Eine Cake-Cutting-Instanz besteht aus einem Kuchen $[0, 1]$, Spielern $N = \{1, \ldots, n\}$

Aufteilung teilbarer Güter



Situation: inhomogener Kuchen

Ziel: jeder bekommt eine faire Portion

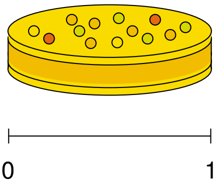
Definition 1

Eine Cake-Cutting-Instanz besteht aus einem Kuchen $[0,1]$, Spielern $N = \{1, \ldots, n\}$ und einer Präferenzfunktion $v_i : \{X \mid X \subseteq [0,1]\} \to \mathbb{R}_{\geq 0}$ für alle $i \in N$, sodass

Aufteilung teilbarer Güter

Situation: inhomogener Kuchen

Ziel: jeder bekommt eine faire Portion



Definition 1

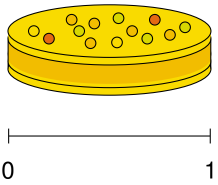
Eine Cake-Cutting-Instanz besteht aus einem Kuchen $[0,1]$, Spielern $N = \{1, \ldots, n\}$ und einer Präferenzfunktion $v_i : \{X \mid X \subseteq [0,1]\} \to \mathbb{R}_{\geq 0}$ für alle $i \in N$, sodass

- $v_i([a,a]) = 0 \ \forall a \in [0,1]$ (Nicht-Atomarität);
- $v_i([0,1]) = 1$ (Normalisierung);
- für jedes Teilintervall $X \subseteq [0,1]$: $v_i(X) \geq 0$ (Nicht-Negativität);

Aufteilung teilbarer Güter

Situation: inhomogener Kuchen

Ziel: jeder bekommt eine faire Portion



Definition 1

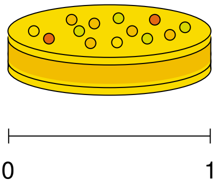
Eine Cake-Cutting-Instanz besteht aus einem Kuchen $[0, 1]$, Spielern $N = \{1, \ldots, n\}$ und einer Präferenzfunktion $v_i : \{X \mid X \subseteq [0, 1]\} \to \mathbb{R}_{\geq 0}$ für alle $i \in N$, sodass

- $v_i([a, a]) = 0 \ \forall a \in [0, 1]$ (Nicht-Atomarität);
- $v_i([0, 1]) = 1$ (Normalisierung);
- für jedes Teilintervall $X \subseteq [0, 1]$: $v_i(X) \geq 0$ (Nicht-Negativität);
- für jedes Teilintervall $[a, b] \subseteq [0, 1]$ und λ: $0 \leq \lambda \leq 1$, existiert ein Punkt $c \in [a, b]$ mit $v_i([a, c]) = \lambda v_i([a, b])$ (Teilbarkeit);

Aufteilung teilbarer Güter

Situation: inhomogener Kuchen

Ziel: jeder bekommt eine faire Portion



Definition 1

Eine Cake-Cutting-Instanz besteht aus einem Kuchen $[0,1]$, Spielern $N = \{1, \ldots, n\}$ und einer Präferenzfunktion $v_i : \{X \mid X \subseteq [0,1]\} \to \mathbb{R}_{\geq 0}$ für alle $i \in N$, sodass

- $v_i([a,a]) = 0 \; \forall a \in [0,1]$ (Nicht-Atomarität);
- $v_i([0,1]) = 1$ (Normalisierung);
- für jedes Teilintervall $X \subseteq [0,1]$: $v_i(X) \geq 0$ (Nicht-Negativität);
- für jedes Teilintervall $[a,b] \subseteq [0,1]$ und λ: $0 \leq \lambda \leq 1$, existiert ein Punkt $c \in [a,b]$ mit $v_i([a,c]) = \lambda v_i([a,b])$ (Teilbarkeit);
- für zwei Teilintervalle $X, Y \subseteq [0,1]$: $v_i(X) + v_i(Y) = v_i(X \cup Y)$ (Additivität).

Aufteilung teilbarer Güter

Beispiel: Cut & Choose

- $n = 2$

Aufteilung teilbarer Güter

Beispiel: Cut & Choose

- $n = 2$
- Spieler 1 schneidet den Kuchen in zwei disjunkte Stücke X_1 und X_2, sodass $v_1(X_1) = v_1(X_2)$ und $X_1 \cup X_2 = [0, 1]$.

Aufteilung teilbarer Güter

Beispiel: Cut & Choose

- $n = 2$
- Spieler 1 schneidet den Kuchen in zwei disjunkte Stücke X_1 und X_2, sodass $v_1(X_1) = v_1(X_2)$ und $X_1 \cup X_2 = [0, 1]$.
- Spieler 2 wählt eine der beiden Stücke X_i mit $v_2(X_i) \geq v_2(X, j)$, $j \neq i$.

Aufteilung teilbarer Güter

Beispiel: Cut & Choose

- $n = 2$
- Spieler 1 schneidet den Kuchen in zwei disjunkte Stücke X_1 und X_2, sodass $v_1(X_1) = v_1(X_2)$ und $X_1 \cup X_2 = [0, 1]$.
- Spieler 2 wählt eine der beiden Stücke X_i mit $v_2(X_i) \geq v_2(X, j)$, $j \neq i$.
- Spieler 1 bekommt das andere Stück.

Aufteilung teilbarer Güter

Beispiel: Cut & Choose

- $n = 2$
- Spieler 1 schneidet den Kuchen in zwei disjunkte Stücke X_1 und X_2, sodass $v_1(X_1) = v_1(X_2)$ und $X_1 \cup X_2 = [0, 1]$.
- Spieler 2 wählt eine der beiden Stücke X_i mit $v_2(X_i) \geq v_2(X, j)$, $j \neq i$.
- Spieler 1 bekommt das andere Stück.

weitere Protokolle, z. B. Dubins–Spanier, Evan–Paz, Selfridge–Conway

Aufteilung teilbarer Güter

Beispiel: Cut & Choose

- $n = 2$
- Spieler 1 schneidet den Kuchen in zwei disjunkte Stücke X_1 und X_2, sodass $v_1(X_1) = v_1(X_2)$ und $X_1 \cup X_2 = [0, 1]$.
- Spieler 2 wählt eine der beiden Stücke X_i mit $v_2(X_i) \geq v_2(X, j)$, $j \neq i$.
- Spieler 1 bekommt das andere Stück.

weitere Protokolle, z. B. Dubins–Spanier, Evan–Paz, Selfridge–Conway

Fairness-Kriterien

- Proportionalität
- Neidfreiheit

Aufteilung teilbarer Güter

Beispiel: Cut & Choose

- $n = 2$
- Spieler 1 schneidet den Kuchen in zwei disjunkte Stücke X_1 und X_2, sodass $v_1(X_1) = v_1(X_2)$ und $X_1 \cup X_2 = [0, 1]$.
- Spieler 2 wählt eine der beiden Stücke X_i mit $v_2(X_i) \geq v_2(X, j)$, $j \neq i$.
- Spieler 1 bekommt das andere Stück.

weitere Protokolle, z. B. Dubins–Spanier, Evan–Paz, Selfridge–Conway

Fairness-Kriterien

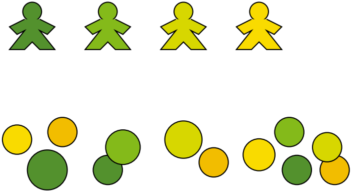
- Proportionalität
- Neidfreiheit

Thema 1

Thema 2

Aufteilung unteilbarer Güter

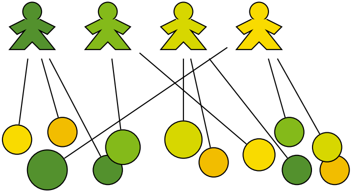
Situation: Objekte/Ressourcen mit unterschiedlichem Wert für Agenten



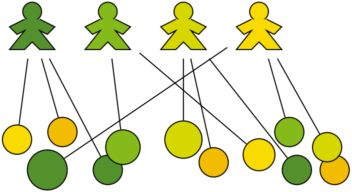
Aufteilung unteilbarer Güter

Situation: Objekte/Ressourcen mit unterschiedlichem Wert für Agenten

Ziel: jeder bekommt einen optimalen Anteil (1 : *many*)



Aufteilung unteilbarer Güter



Situation: Objekte/Ressourcen mit unterschiedlichem Wert für Agenten

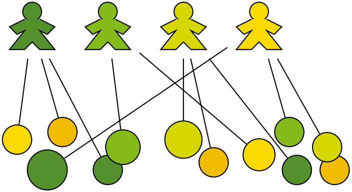
Ziel: jeder bekommt einen optimalen Anteil (1 : *many*)

Definition 2 (*MultiAgent Resource Allocation* (MARA))

Eine MARA-Instanz besteht aus

- Agenten $N = \{1, \ldots, n\}$
- Objekten $\mathcal{O} = \{o_1, \ldots, o_p\}$
- Präferenzen der Agenten in N

Aufteilung unteilbarer Güter



Situation: Objekte/Ressourcen mit unterschiedlichem Wert für Agenten

Ziel: jeder bekommt einen optimalen Anteil (1 : *many*)

Definition 2 (*MultiAgent Resource Allocation* (MARA))

Eine MARA-Instanz besteht aus

- Agenten $N = \{1, \dots, n\}$
- Objekten $\mathcal{O} = \{o_1, \dots, o_p\}$
- Präferenzen der Agenten in N

unterschiedliche Darstellungen der Präferenzen:

- kardinale Präferenzen (Nutzenfunktion $u : \mathcal{O} \to \mathbb{R}$)
- ordinale Präferenzen (z. B. Ranking über Bündel von Objekten)

Aufteilung unteilbarer Güter

Beispiel: das *Santa-Claus*-Problem

- MARA-Instanz mit kardinalen Präferenzen
- Nutzenfunktion u ist modular: für alle $S, T \subseteq \mathcal{O}$:
  $u(S \cup T) = u(S) + u(T) - u(S \cap T)$.
- Ziel: Nutzen für das unglücklichste Kind maximieren.

Aufteilung unteilbarer Güter

Beispiel: das *Santa-Claus*-Problem

- MARA-Instanz mit kardinalen Präferenzen
- Nutzenfunktion u ist modular: für alle $S, T \subseteq \mathcal{O}$:
  $u(S \cup T) = u(S) + u(T) - u(S \cap T)$.
- Ziel: Nutzen für das unglücklichste Kind maximieren.

Fairness vs. Effizienz:

- Maxmin-Aufteilung
- Neidfreiheit
- Pareto-Effizienz

Aufteilung unteilbarer Güter

Beispiel: das *Santa-Claus*-Problem

- MARA-Instanz mit kardinalen Präferenzen
- Nutzenfunktion u ist modular: für alle $S, T \subseteq \mathcal{O}$:
  $u(S \cup T) = u(S) + u(T) - u(S \cap T)$.
- Ziel: Nutzen für das unglücklichste Kind maximieren.

Fairness vs. Effizienz:

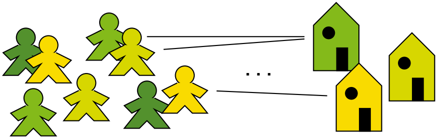
- Maxmin-Aufteilung
- Neidfreiheit
- Pareto-Effizienz

Thema 3

Zuweisung von Ärzten auf Krankenhäuser

Situation: beidseitige Präferenzen

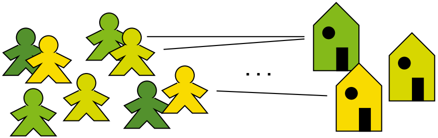
Ziel: (*many* : 1)-Zuweisung



Zuweisung von Ärzten auf Krankenhäuser

Situation: beidseitige Präferenzen

Ziel: (*many* : 1)-Zuweisung



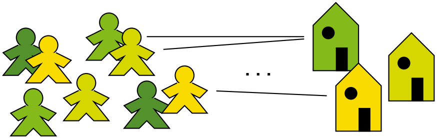
Definition 3 (Das Hospital-Residents-Problem (HR))

Eine HR-Instanz besteht aus

Zuweisung von Ärzten auf Krankenhäuser

Situation: beidseitige Präferenzen

Ziel: (*many* : 1)-Zuweisung



Definition 3 (Das Hospital-Residents-Problem (HR))

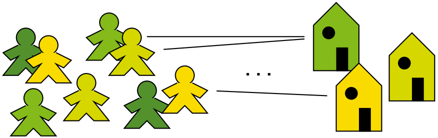
Eine HR-Instanz besteht aus

- den disjunkten Agentenmengen $R = \{r_1, \dots, r_{n_1}\}$ und $H = \{h_1, \dots, h_{n_2}\}$

Zuweisung von Ärzten auf Krankenhäuser

Situation: beidseitige Präferenzen

Ziel: (*many* : 1)-Zuweisung



Definition 3 (Das Hospital-Residents-Problem (HR))

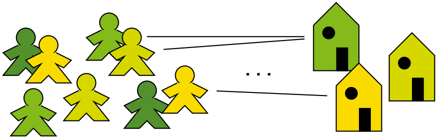
Eine HR-Instanz besteht aus

- den disjunkten Agentenmengen $R = \{r_1, \ldots, r_{n_1}\}$ und $H = \{h_1, \ldots, h_{n_2}\}$
- $\forall h_j \in H$: Kapazität c_j

Zuweisung von Ärzten auf Krankenhäuser

Situation: beidseitige Präferenzen

Ziel: (*many* : 1)-Zuweisung



Definition 3 (Das Hospital-Residents-Problem (HR))

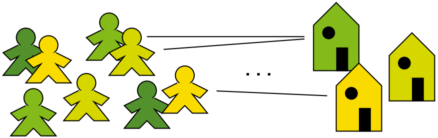
Eine HR-Instanz besteht aus

- den disjunkten Agentenmengen $R = \{r_1, \ldots, r_{n_1}\}$ und $H = \{h_1, \ldots, h_{n_2}\}$
- $\forall h_j \in H$: Kapazität c_j
- $E \subseteq R \times H$ akzeptable Paare, $m = \|E\|$
- $\forall r_i \in R$: $A(r_i) = \{h_j \in H \mid (r_i, h_j) \in E\}$, $\forall h_j \in R$: $A(h_j) = \{r_i \in R \mid (r_i, h_j) \in E\}$

Zuweisung von Ärzten auf Krankenhäuser

Situation: beidseitige Präferenzen

Ziel: (*many* : 1)-Zuweisung



Definition 3 (Das Hospital-Residents-Problem (HR))

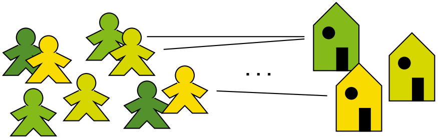
Eine HR-Instanz besteht aus

- den disjunkten Agentenmengen $R = \{r_1, \ldots, r_{n_1}\}$ und $H = \{h_1, \ldots, h_{n_2}\}$
- $\forall h_j \in H$: Kapazität c_j
- $E \subseteq R \times H$ akzeptable Paare, $m = \|E\|$
- $\forall r_i \in R$: $A(r_i) = \{h_j \in H \mid (r_i, h_j) \in E\}$, $\quad \forall h_j \in R$: $A(h_j) = \{r_i \in R \mid (r_i, h_j) \in E\}$
- $\forall a_k \in R \cup H$: strikte Präferenzliste über $A(k)$

Zuweisung von Ärzten auf Krankenhäuser

Situation: beidseitige Präferenzen

Ziel: (*many* : 1)-Zuweisung



Definition 3 (Das Hospital-Residents-Problem (HR))

Eine HR-Instanz besteht aus

- den disjunkten Agentenmengen $R = \{r_1, \dots, r_{n_1}\}$ und $H = \{h_1, \dots, h_{n_2}\}$
- $\forall h_j \in H$: Kapazität c_j
- $E \subseteq R \times H$ akzeptable Paare, $m = \|E\|$
- $\forall r_i \in R$: $A(r_i) = \{h_j \in H \mid (r_i, h_j) \in E\}$, $\forall h_j \in R$: $A(h_j) = \{r_i \in R \mid (r_i, h_j) \in E\}$
- $\forall a_k \in R \cup H$: strikte Präferenzliste über $A(k)$

M ist eine Zuweisung (*Matching*), wenn $|M(r_i)| \leq 1 \; \forall r_i \in R$ und $|M(h_j)| \leq c_j \; \forall h_j \in H$

Zuweisung von Ärzten auf Krankenhäuser

Seien I eine HR-Instanz und M eine Zuweisung in I.

$(r_i, h_j) \in E \setminus M$ blockiert M, falls

Zuweisung von Ärzten auf Krankenhäuser

Seien I eine HR-Instanz und M eine Zuweisung in I.

$(r_i, h_j) \in E \setminus M$ blockiert M, falls

- Arzt r_i ist nicht zugewiesen oder bevorzugt h_j vor $M(r_i)$;

Zuweisung von Ärzten auf Krankenhäuser

Seien I eine HR-Instanz und M eine Zuweisung in I.

$(r_i, h_j) \in E \setminus M$ blockiert M, falls

- Arzt r_i ist nicht zugewiesen oder bevorzugt h_j vor $M(r_i)$;
- Krankenhaus h_j ist unterbesetzt oder bevorzugt r_i vor mindestens einem Arzt in $M(h_j)$.

Zuweisung von Ärzten auf Krankenhäuser

Seien I eine HR-Instanz und M eine Zuweisung in I.

$(r_i, h_j) \in E \setminus M$ blockiert M, falls

- Arzt r_i ist nicht zugewiesen oder bevorzugt h_j vor $M(r_i)$;
- Krankenhaus h_j ist unterbesetzt oder bevorzugt r_i vor mindestens einem Arzt in $M(h_j)$.

Eine Zuweisung M heißt stabil, falls kein Paar M blockiert.

Zuweisung von Ärzten auf Krankenhäuser

Seien I eine HR-Instanz und M eine Zuweisung in I.

$(r_i, h_j) \in E \setminus M$ blockiert M, falls

- Arzt r_i ist nicht zugewiesen oder bevorzugt h_j vor $M(r_i)$;
- Krankenhaus h_j ist unterbesetzt oder bevorzugt r_i vor mindestens einem Arzt in $M(h_j)$.

Eine Zuweisung M heißt stabil, falls kein Paar M blockiert.

Satz 4

Es gibt immer eine stabile Zuweisung.

Zuweisung von Ärzten auf Krankenhäuser

Gale–Shapley-Algorithmus (*resident oriented*, 1962)

Eingabe: HR-Instanz

Ausgabe: stabile Zuweisung M (optimal für Ärzte)

Zuweisung von Ärzten auf Krankenhäuser

Gale–Shapley-Algorithmus (*resident oriented*, 1962)

Nobelpreis an L. Shapley und A. Roth, 2012

Eingabe: HR-Instanz

Ausgabe: stabile Zuweisung M (optimal für Ärzte)

Zuweisung von Ärzten auf Krankenhäuser

Nobelpreis an L. Shapley und A. Roth, 2012

Gale–Shapley-Algorithmus (*resident oriented*, 1962)

Eingabe: HR-Instanz
Ausgabe: stabile Zuweisung M (optimal für Ärzte)

1. alle Ärzte frei
2. alle Krankenhäuser komplett unbesetzt

Zuweisung von Ärzten auf Krankenhäuser

Nobelpreis an L. Shapley und A. Roth, 2012

Gale–Shapley-Algorithmus (*resident oriented*, 1962)

Eingabe: HR-Instanz

Ausgabe: stabile Zuweisung M (optimal für Ärzte)

1. alle Ärzte frei
2. alle Krankenhäuser komplett unbesetzt
3. Solange (ein $r \in R$ frei) und (Präferenzliste von r nicht-leer):
4. $\quad h \leftarrow$ erstes Krankenhaus auf Liste von r

Zuweisung von Ärzten auf Krankenhäuser

Nobelpreis an L. Shapley und A. Roth, 2012

Gale–Shapley-Algorithmus (*resident oriented*, 1962)

Eingabe: HR-Instanz

Ausgabe: stabile Zuweisung M (optimal für Ärzte)

```
alle Ärzte frei
alle Krankenhäuser komplett unbesetzt
Solange (ein r ∈ R frei) und (Präferenzliste von r nicht-leer):
    h ← erstes Krankenhaus auf Liste von r
    Falls h vollständig besetzt:
        r′ ← schlechtester Arzt, der bisher h zugewiesen ist
        r′ frei
```

Zuweisung von Ärzten auf Krankenhäuser

Nobelpreis an L. Shapley und A. Roth, 2012

Gale–Shapley-Algorithmus (*resident oriented*, 1962)

Eingabe: HR-Instanz

Ausgabe: stabile Zuweisung M (optimal für Ärzte)

alle Ärzte frei
alle Krankenhäuser komplett unbesetzt
Solange (ein $r \in R$ frei) und (Präferenzliste von r nicht-leer):
    $h \leftarrow$ erstes Krankenhaus auf Liste von r
    Falls h vollständig besetzt:
        $r' \leftarrow$ schlechtester Arzt, der bisher h zugewiesen ist
        r' frei
    Weise r zu h zu

Zuweisung von Ärzten auf Krankenhäuser

Nobelpreis an L. Shapley und A. Roth, 2012

Gale–Shapley-Algorithmus (*resident oriented*, 1962)

Eingabe: HR-Instanz

Ausgabe: stabile Zuweisung M (optimal für Ärzte)

```
alle Ärzte frei
alle Krankenhäuser komplett unbesetzt
Solange (ein r ∈ R frei) und (Präferenzliste von r nicht-leer):
        h ← erstes Krankenhaus auf Liste von r
        Falls h vollständig besetzt:
                r′ ← schlechtester Arzt, der bisher h zugewiesen ist
                r′ frei
        Weise r zu h zu
        Falls h vollständig besetzt:
                s′ ← schlechtester Arzt, der bisher h zugewiesen ist
```

Zuweisung von Ärzten auf Krankenhäuser

Nobelpreis an L. Shapley und A. Roth, 2012

Gale–Shapley-Algorithmus (*resident oriented*, 1962)

Eingabe: HR-Instanz

Ausgabe: stabile Zuweisung M (optimal für Ärzte)

```
alle Ärzte frei
alle Krankenhäuser komplett unbesetzt
Solange (ein r ∈ R frei) und (Präferenzliste von r nicht-leer):
    h ← erstes Krankenhaus auf Liste von r
    Falls h vollständig besetzt:
        r′ ← schlechtester Arzt, der bisher h zugewiesen ist
        r′ frei
    Weise r zu h zu
    Falls h vollständig besetzt:
        s′ ← schlechtester Arzt, der bisher h zugewiesen ist
        Für jeden Nachfolger s′ von s auf Liste von h
            Entferne s′ und h aus deren jeweiligen Listen
```

Zuweisung von Ärzten auf Krankenhäuser

Gale–Shapley-Algorithmus (*resident oriented*, 1962)

Nobelpreis an L. Shapley und A. Roth, 2012

Eingabe: HR-Instanz
Ausgabe: stabile Zuweisung M (optimal für Ärzte)

```
alle Ärzte frei
alle Krankenhäuser komplett unbesetzt
Solange (ein r ∈ R frei) und (Präferenzliste von r nicht-leer):
    h ← erstes Krankenhaus auf Liste von r
    Falls h vollständig besetzt:
        r′ ← schlechtester Arzt, der bisher h zugewiesen ist
        r′ frei
    Weise r zu h zu
    Falls h vollständig besetzt:
        s′ ← schlechtester Arzt, der bisher h zugewiesen ist
        Für jeden Nachfolger s′ von s auf Liste von h
            Entferne s′ und h aus deren jeweiligen Listen
Gib die Menge der zugewiesenen Paare zurück.
```

Zuweisung von Ärzten auf Krankenhäuser

Der oben angegebene Algorithmus gibt die eindeutige stabile Zuweisung M_a aus, die aus Sicht der Ärzte optimal ist.

Zuweisung von Ärzten auf Krankenhäuser

Der oben angegebene Algorithmus gibt die eindeutige stabile Zuweisung M_a aus, die aus Sicht der Ärzte optimal ist.

Analog gibt es eine eindeutige stabile Zuweisung M_z, die aus Sicht der Krankenhäuser optimal ist.

Zuweisung von Ärzten auf Krankenhäuser

Der oben angegebene Algorithmus gibt die eindeutige stabile Zuweisung M_a aus, die aus Sicht der Ärzte optimal ist.

Analog gibt es eine eindeutige stabile Zuweisung M_z, die aus Sicht der Krankenhäuser optimal ist.

Dazwischen kann es noch weitere stabile Zuweisungen geben, die jedes Krankenhaus mindestens so gut findet wie M_a und jeder Arzt mindestens so gut wie M_z.

Zuweisung von Ärzten auf Krankenhäuser

Der oben angegebene Algorithmus gibt die eindeutige stabile Zuweisung M_a aus, die aus Sicht der Ärzte optimal ist.

Analog gibt es eine eindeutige stabile Zuweisung M_z, die aus Sicht der Krankenhäuser optimal ist.

Dazwischen kann es noch weitere stabile Zuweisungen geben, die jedes Krankenhaus mindestens so gut findet wie M_a und jeder Arzt mindestens so gut wie M_z.

Satz 5 (*Rural Hospitals*)

Für eine gegebene HR*-Instanz gelten:*

Zuweisung von Ärzten auf Krankenhäuser

Der oben angegebene Algorithmus gibt die eindeutige stabile Zuweisung M_a aus, die aus Sicht der Ärzte optimal ist.

Analog gibt es eine eindeutige stabile Zuweisung M_z, die aus Sicht der Krankenhäuser optimal ist.

Dazwischen kann es noch weitere stabile Zuweisungen geben, die jedes Krankenhaus mindestens so gut findet wie M_a und jeder Arzt mindestens so gut wie M_z.

Satz 5 (*Rural Hospitals*)

Für eine gegebene HR*-Instanz gelten:*

- *In allen stabilen Zuweisungen bekommen die gleichen Ärzte ein Krankenhaus zugewiesen.*

Zuweisung von Ärzten auf Krankenhäuser

Der oben angegebene Algorithmus gibt die eindeutige stabile Zuweisung M_a aus, die aus Sicht der Ärzte optimal ist.

Analog gibt es eine eindeutige stabile Zuweisung M_z, die aus Sicht der Krankenhäuser optimal ist.

Dazwischen kann es noch weitere stabile Zuweisungen geben, die jedes Krankenhaus mindestens so gut findet wie M_a und jeder Arzt mindestens so gut wie M_z.

Satz 5 (*Rural Hospitals*)

Für eine gegebene HR*-Instanz gelten:*

- *In allen stabilen Zuweisungen bekommen die gleichen Ärzte ein Krankenhaus zugewiesen.*
- *Jedes Krankenhaus bekommt in allen stabilen Zuweisungen die gleiche Anzahl an Ärzten.*

Zuweisung von Ärzten auf Krankenhäuser

Der oben angegebene Algorithmus gibt die eindeutige stabile Zuweisung M_a aus, die aus Sicht der Ärzte optimal ist.

Analog gibt es eine eindeutige stabile Zuweisung M_z, die aus Sicht der Krankenhäuser optimal ist.

Dazwischen kann es noch weitere stabile Zuweisungen geben, die jedes Krankenhaus mindestens so gut findet wie M_a und jeder Arzt mindestens so gut wie M_z.

Satz 5 (*Rural Hospitals*)

Für eine gegebene HR*-Instanz gelten:*

- *In allen stabilen Zuweisungen bekommen die gleichen Ärzte ein Krankenhaus zugewiesen.*
- *Jedes Krankenhaus bekommt in allen stabilen Zuweisungen die gleiche Anzahl an Ärzten.*
- *Jedes Krankenhaus, das in einer stabilen Zuweisung unterbesetzt ist, bekommt in jeder stabilen Zuweisung die gleichen Ärzten zugewiesen.*

Zuweisung von Ärzten auf Krankenhäuser

Varianten:

- stabiles Heiratsproblem (1:1)

Zuweisung von Ärzten auf Krankenhäuser

Varianten:

- stabiles Heiratsproblem (1:1)
- Indifferenzen in Präferenzen: schwache, starke und Super-Stabilität

Zuweisung von Ärzten auf Krankenhäuser

Varianten:

Thema 5

Thema 6

- stabiles Heiratsproblem (1:1)
- Indifferenzen in Präferenzen: schwache, starke und Super-Stabilität

Zuweisung von Ärzten auf Krankenhäuser

Varianten:

- stabiles Heiratsproblem (1:1)
- Indifferenzen in Präferenzen: schwache, starke und Super-Stabilität
- Berücksichtigung von Paaren

Thema 5

Thema 6

Thema 9

Zuweisung von Ärzten auf Krankenhäuser

Varianten:

- stabiles Heiratsproblem (1:1)
- Indifferenzen in Präferenzen: schwache, starke und Super-Stabilität
- Berücksichtigung von Paaren
- Quoten: Mindestanzahlen und gemeinsame Kapazitäten

Thema 5

Thema 6

Thema 9

Thema 10

Zuweisung von Ärzten auf Krankenhäuser

Varianten:

- stabiles Heiratsproblem (1:1)
- Indifferenzen in Präferenzen: schwache, starke und Super-Stabilität (Thema 5, Thema 6)
- Berücksichtigung von Paaren (Thema 9)
- Quoten: Mindestanzahlen und gemeinsame Kapazitäten (Thema 10)
- (many:many)-Varianten: Zuweisung von Arbeitern und Firmen

Zuweisung von Ärzten auf Krankenhäuser

Varianten:

Thema 5

Thema 6

- stabiles Heiratsproblem (1:1)
- Indifferenzen in Präferenzen: schwache, starke und Super-Stabilität
- Berücksichtigung von Paaren — Thema 9
- Quoten: Mindestanzahlen und gemeinsame Kapazitäten — Thema 10
- (many:many)-Varianten: Zuweisung von Arbeitern und Firmen
- das Projekt-Zuweisungs-Problem — Thema 11
  - Kapzitäten von Projekten und Dozenten
  - Studierende und Dozenten haben Präferenzen über Teilmengen (Studierende)
  - Angebote von Dozenten

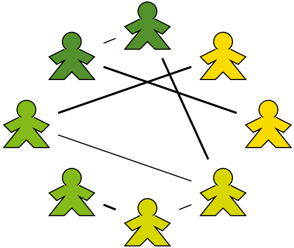
Zuweisung von Ärzten auf Krankenhäuser

Varianten:

Thema 5 Thema 6

- stabiles Heiratsproblem (1:1)
- Indifferenzen in Präferenzen: schwache, starke und Super-Stabilität
- Berücksichtigung von Paaren Thema 9
- Quoten: Mindestanzahlen und gemeinsame Kapazitäten Thema 10
- (many:many)-Varianten: Zuweisung von Arbeitern und Firmen
- das Projekt-Zuweisungs-Problem Thema 11
  - Kapzitäten von Projekten und Dozenten
  - Studierende und Dozenten haben Präferenzen über Teilmengen (Studierende)
  - Angebote von Dozenten
- tripartite Zuweisung
- ...

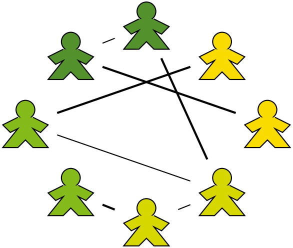
Stabiles Mitbewohnerproblem



Situation: nicht-bipartite, möglicherweise unvollständige Präferenzen über alle möglichen Mitbewohner

Ziel: paarweise Zuweisung

Stabiles Mitbewohnerproblem



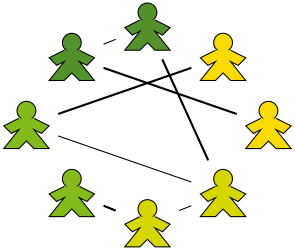
Situation: nicht-bipartite, möglicherweise unvollständige Präferenzen über alle möglichen Mitbewohner

Ziel: paarweise Zuweisung

Definition 6 (Das Stable-Roommates-Problem (SR))

Eine SR-Instanz besteht aus

Stabiles Mitbewohnerproblem



Situation: nicht-bipartite, möglicherweise unvollständige Präferenzen über alle möglichen Mitbewohner

Ziel: paarweise Zuweisung

Definition 6 (Das Stable-Roommates-Problem (SR))

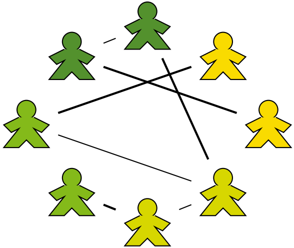
Eine SR-Instanz besteht aus

- Agenten $A = \{a_1, \ldots, a_n\}$

Stabiles Mitbewohnerproblem

Situation: nicht-bipartite, möglicherweise unvollständige Präferenzen über alle möglichen Mitbewohner

Ziel: paarweise Zuweisung



Definition 6 (Das Stable-Roommates-Problem (SR))

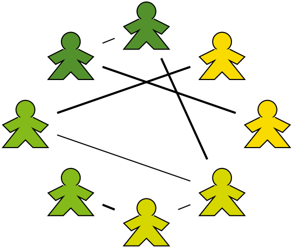
Eine SR-Instanz besteht aus

- Agenten $A = \{a_1, \ldots, a_n\}$
- $E = \{\{a_i, a_j\} \mid a_i, a_j \in A, i \neq j, \{a_i, a_j\}$ akzeptable Paare$\}$, $m = \|E\|$
- $\forall a_i \in A$: $A(a_i) = \{a_j \in E \mid \{a_i, a_j\} \in E\}$

Stabiles Mitbewohnerproblem

Situation: nicht-bipartite, möglicherweise unvollständige Präferenzen über alle möglichen Mitbewohner

Ziel: paarweise Zuweisung



Definition 6 (Das Stable-Roommates-Problem (SR))

Eine SR-Instanz besteht aus

- Agenten $A = \{a_1, \ldots, a_n\}$
- $E = \{\{a_i, a_j\} \mid a_i, a_j \in A, i \neq j, \{a_i, a_j\}$ akzeptable Paare$\}$, $m = \|E\|$
- $\forall a_i \in A$: $A(a_i) = \{a_j \in E \mid \{a_i, a_j\} \in E\}$
- $\forall a_i \in A$: strikte Präferenzliste über $A(a_i)$

Stabiles Mitbewohnerproblem

Sei I eine SR-Instanz. $M \subseteq E$ ist eine Zuweisung in I, falls kein Agent zu mehr als einem anderen Agenten zugeordnet wird.

Stabiles Mitbewohnerproblem

Sei I eine SR-Instanz. $M \subseteq E$ ist eine Zuweisung in I, falls kein Agent zu mehr als einem anderen Agenten zugeordnet wird.

Ein Paar $(a_i, a_j) \in E \setminus M$ blockiert M, falls

- Agent a_i nicht zugewiesen ist oder a_j vor $M(a_i)$ bevorzugt;
- Agent a_j nicht zugewiesen ist oder a_i vor $M(a_j)$ bevorzugt.

Stabiles Mitbewohnerproblem

Sei I eine SR-Instanz. $M \subseteq E$ ist eine Zuweisung in I, falls kein Agent zu mehr als einem anderen Agenten zugeordnet wird.

Ein Paar $(a_i, a_j) \in E \setminus M$ blockiert M, falls

- Agent a_i nicht zugewiesen ist oder a_j vor $M(a_i)$ bevorzugt;
- Agent a_j nicht zugewiesen ist oder a_i vor $M(a_j)$ bevorzugt.

M heißt stabil, falls kein Paar M blockiert.

Stabiles Mitbewohnerproblem

Sei I eine SR-Instanz. $M \subseteq E$ ist eine Zuweisung in I, falls kein Agent zu mehr als einem anderen Agenten zugeordnet wird.

Ein Paar $(a_i, a_j) \in E \setminus M$ blockiert M, falls

- Agent a_i nicht zugewiesen ist oder a_j vor $M(a_i)$ bevorzugt;
- Agent a_j nicht zugewiesen ist oder a_i vor $M(a_j)$ bevorzugt.

M heißt stabil, falls kein Paar M blockiert.

I heißt lösbar, falls es eine stabile Zuweisung in I gibt.

Stabiles Mitbewohnerproblem

Sei I eine SR-Instanz. $M \subseteq E$ ist eine Zuweisung in I, falls kein Agent zu mehr als einem anderen Agenten zugeordnet wird.

Ein Paar $(a_i, a_j) \in E \setminus M$ blockiert M, falls

- Agent a_i nicht zugewiesen ist oder a_j vor $M(a_i)$ bevorzugt;
- Agent a_j nicht zugewiesen ist oder a_i vor $M(a_j)$ bevorzugt.

M heißt stabil, falls kein Paar M blockiert.

I heißt lösbar, falls es eine stabile Zuweisung in I gibt.

Konzepte

- SR-Instanzen sind nicht immer lösbar.
- Es gibt immer eine stabile Partitionierung.
- fast-stabile Zuweisungen berücksichtigen geringe Kompromisse.

Stabiles Mitbewohnerproblem

Sei I eine SR-Instanz. $M \subseteq E$ ist eine Zuweisung in I, falls kein Agent zu mehr als einem anderen Agenten zugeordnet wird.

Ein Paar $(a_i, a_j) \in E \setminus M$ blockiert M, falls

- Agent a_i nicht zugewiesen ist oder a_j vor $M(a_i)$ bevorzugt;
- Agent a_j nicht zugewiesen ist oder a_i vor $M(a_j)$ bevorzugt.

M heißt stabil, falls kein Paar M blockiert.

I heißt lösbar, falls es eine stabile Zuweisung in I gibt.

Konzepte

- SR-Instanzen sind nicht immer lösbar.
- Es gibt immer eine stabile Partitionierung.
- fast-stabile Zuweisungen berücksichtigen geringe Kompromisse.

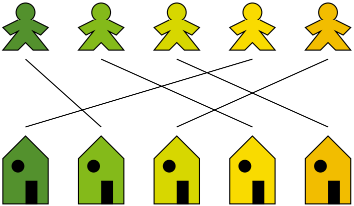
Thema 7

Thema 8

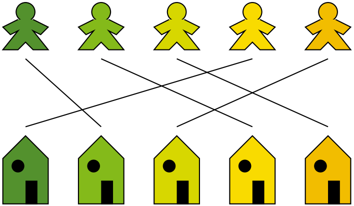
Zuweisung von Häusern

Situation: einseitige Präferenzen von Agenten über Häuser

Ziel: (1 : 1)-Zuweisung



Zuweisung von Häusern



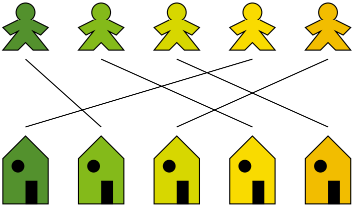
Situation: einseitige Präferenzen von Agenten über Häuser

Ziel: (1 : 1)-Zuweisung

Optimalitätskriterien

- Pareto-Optimalität
- Popularität
- profilbasierte Optimalität

Zuweisung von Häusern



Situation: einseitige Präferenzen von Agenten über Häuser

Ziel: (1 : 1)-Zuweisung

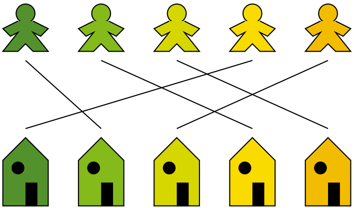
Optimalitätskriterien

- Pareto-Optimalität
- Popularität
- profilbasierte Optimalität

Varianten:

- *Housing Markets*: bestehende Zuweisung, individuelle Rationalität
- Gleichstände
- (*many*:1)-Verallgemeinerung: WG mit Kapazitäten

Zuweisung von Häusern



Situation: einseitige Präferenzen von Agenten über Häuser

Ziel: $(1:1)$-Zuweisung

Optimalitätskriterien

- Pareto-Optimalität
- Popularität
- profilbasierte Optimalität

Thema 4

Thema 12

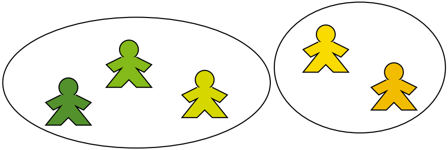
Thema 13

Thema 14

Varianten:

- *Housing Markets*: bestehende Zuweisung, individuelle Rationalität
- Gleichstände
- (*many*:1)-Verallgemeinerung: WG mit Kapazitäten

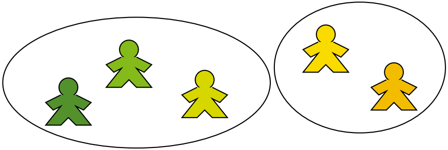
hedonische Spiele



Situation: Koalitionsbildung; Zufriedenheit der Spieler hängt nur von eigener Koalition ab.

hedonische Spiele

Situation: Koalitionsbildung; Zufriedenheit der Spieler hängt nur von eigener Koalition ab.



Definition 7

Ein hedonisches Spiel $(N, \succeq)$ besteht aus:

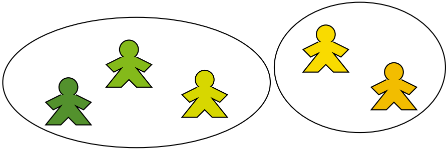
- endlicher Spielermenge $N = \{1, \ldots, n\}$,
- Präferenzprofil $\succeq = (\succeq_1, \ldots, \succeq_n)$
  mit $\succeq_i$ totaler Präferenzrelation über $\mathcal{N}_i = \{C \subseteq N \mid i \in C\}$ für $i \in N$.

Also ist $\succeq$ reflexiv, transitiv, nicht notwendigerweise antisymmetrisch und total.

Eine Teilmenge $C \subseteq N$ heißt Koalition.
Ein Partitionierung Γ von N heißt Koalitionsstruktur. $\Gamma(i) \in \Gamma$ mit $i \in \Gamma(i)$.

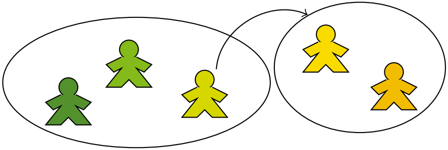
hedonische Spiele

Ziel: stabile Koalitionsstruktur finden



hedonische Spiele

Ziel: stabile Koalitionsstruktur finden



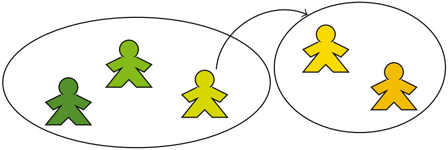
Definition 8

Sei ein hedonisches Spiel $(N, \succeq)$. Eine Koalitionsstruktur heißt

- Nash-stabil, wenn $\forall i \in N, \forall C \in \Gamma \colon \Gamma(i) \succeq C$.

hedonische Spiele

Ziel: stabile Koalitionsstruktur finden



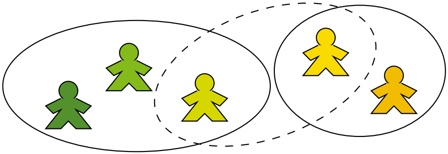
Definition 8

Sei ein hedonisches Spiel $(N, \succeq)$. Eine Koalitionsstruktur heißt

- Nash-stabil, wenn $\forall i \in N$, $\forall C \in \Gamma$: $\Gamma(i) \succeq C$.
- Varianten: individuell stabil und vertraglich individuell stabil

hedonische Spiele

Ziel: stabile Koalitionsstruktur finden

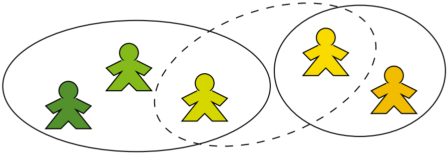


Definition 8

Sei ein hedonisches Spiel $(N, \succeq)$. Eine Koalitionsstruktur heißt

- kernstabil, wenn es keine blockierende Koalition gibt,
  d. h., $\forall C \subseteq N, \exists i \in C\colon \Gamma(i) \geq C$.

hedonische Spiele



Ziel: stabile Koalitionsstruktur finden

Definition 8

Sei ein hedonisches Spiel $(N, \succeq)$. Eine Koalitionsstruktur heißt

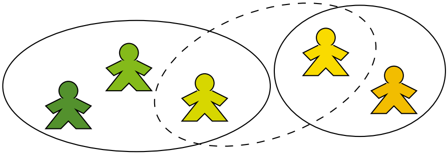
- kernstabil, wenn es keine blockierende Koalition gibt,
  d. h., $\forall C \subseteq N, \exists i \in C\colon \Gamma(i) \succeq C$.

Darstellungen:

- individuell rational
- additiv separabel
- ...

hedonische Spiele

Ziel: stabile Koalitionsstruktur finden



Definition 8

Sei ein hedonisches Spiel $(N, \succeq)$. Eine Koalitionsstruktur heißt

- kernstabil, wenn es keine blockierende Koalition gibt,
  d. h., $\forall C \subseteq N, \exists i \in C \colon \Gamma(i) \geq C$.

Darstellungen:

- individuell rational
- additiv separabel
- ...

Thema 15

Vortragstermine I

	Thema	Datum
1	Cake-Cutting – Proportionalität	14.12.2016
2	Cake-Cutting – Neidfreiheit	14.12.2016
3	Aufteilung unteilbarer Güter	21.12.2016
4	HA – Pareto-Optimaliät	21.12.2016
	— Ferien —	
5	HR mit Indifferenzen – schwache Stabilität	11.01.2017
6	HR mit Indifferenzen – starke & Super-Stabilität	11.01.2017
7	SR – stabile Partitionierungen	18.01.2017
8	SR – fast-stabile Zuweisungen	18.01.2017

Vortragstermine II

	Thema	Datum
9	HR mit Paaren	25.01.2017
10	HR mit Quoten	25.01.2017
11	Das Projekt-Zuweisungs-Problem	01.02.2017
12	HA mit Kapazitäten	01.02.2017
13	HA – Popularität	02.02.2017
14	HA – profilbasierte Optimalität	08.02.2017
15	Koalitionsbildung in hedonischen Spielen	08.02.2017

Literatur

F. Brandt, V. Conitzer, U. Endriss, J. Lang und A. Procaccia, Editoren.
Handbook of Computational Social Choice.
Cambridge University Press, 2016.

D. Gusfield und R. Irving
The Stable Marriage Problem: Structure and Algorithms.
MIT Press, 1989.

D. Knuth.
Stable Marriage and its Relation to Other Combinatorial Problems.
volume 10 of CRM Proceedings and Lecture Notes, American Mathematical Society, 1997. Original: Mariages Stables, Les Presses de L' Université de446 Montreal, 1976.

D. Manlove
Algorithmics Of Matching Under Preferences.
Volume 2 of Theoretical computer science. World Scientific Publishing, 2013.

. . . und jede Menge Referenzen darin.

technische universität
dortmund

Effiziente Algorithmen und
Komplexitätstheorie /

Informatik

Aufteilungs- und Zuweisungsalgorithmen

Proseminar
im Wintersemester 2016/2017

Anja Rey, 26. Oktober 2016

Zeitplan

Beginn

- Mi 19.10.16 Einleitung, Themen
- Do 20.10.16 Beginn Präsentationskurs

Themenvergabe

- Thema auswählen bis 25.10 unter

  `http://doodle.com/poll/37hfa3gxyn4xdc7n`

Ausarbeitungsphase

- Mi 26.10.16 Besprechung
- Mi 02.11.16 & Mi 09.11.16 Fragen & Bearbeitung
- Di 15.11.16 erste Abgabe Ausarbeitung

— Rechtzeitig Vortragsplanung beginnen! —

Reviewphase

- Verteilung Reviews (per E-Mail)
- Mi 16.11.16 Besprechung
- Mi 23.11.16 Fragen & Bearbeitung
- Do 24.11.16 Abgabe Reviews
- Mi 30.11.16 Fragen & Bearbeitung
- Mi 30.11.16 zweite Abgabe Ausarbeitung

Zwischenbesprechung

- 8. Woche Einzeltermine
- Do 08.12.16 Ende Präsentationskurs

Vortragsphase und Ende

- 9.–15. Woche: Vorträge
- Feedback am Ende

Vortragstermine Update I

	Thema	Datum
1	Cake-Cutting – Proportionalität	14.12.2016
2	Cake-Cutting – Neidfreiheit	14.12.2016
3	Aufteilung unteilbarer Güter	21.12.2016
4	HA – Pareto-Optimaliät	21.12.2016
	— Ferien —	
5	HR mit Indifferenzen – schwache Stabilität	11.01.2017
6	HR mit Indifferenzen – starke & Super-Stabilität	11.01.2017
7	SR – stabile Partitionierungen	18.01.2017
8	SR – fast-stabile Zuweisungen	18.01.2017

Vortragstermine Update II

	Thema	Datum
9	HR mit Paaren	25.01.2017
~~10~~	~~HR mit Quoten~~	~~25.01.2017~~
11	Das Projekt-Zuweisungs-Problem	25.01.2017
~~11~~	~~Das Projekt-Zuweisungs-Problem~~	~~01.02.2017~~
12	HA mit Kapazitäten	01.02.2017
~~13~~	~~HA – Popularität~~	~~02.02.2017~~
15	Koalitionsbildung in hedonischen Spielen	01.02.2017
~~14~~	~~HA – profilbasierte Optimalität~~	~~08.02.2017~~
~~15~~	~~Koalitionsbildung in hedonischen Spielen~~	~~08.02.2017~~

Anforderungen Ausarbeitung

Format & Struktur

- 5–7 A4-Seiten, PDF-Format, LaTeX verwenden
- sinnvoll strukturieren (Titel, Einleitung, Grundlagen, Hauptteil, Schluss)

Anforderungen Ausarbeitung

Format & Struktur

- 5–7 A4-Seiten, PDF-Format, LaTeX verwenden
- sinnvoll strukturieren (Titel, Einleitung, Grundlagen, Hauptteil, Schluss)

Inhalt

- Definitionen und Kernaussagen herausfiltern
- in eigenen Worten wiedergeben (reine Übersetzung nicht ausreichend)
- auf eigene Beispiele übertragen, Thema daran veranschaulichen

Anforderungen Ausarbeitung

Format & Struktur

- 5–7 A4-Seiten, PDF-Format, LaTeX verwenden
- sinnvoll strukturieren (Titel, Einleitung, Grundlagen, Hauptteil, Schluss)

Inhalt

- Definitionen und Kernaussagen herausfiltern
- in eigenen Worten wiedergeben (reine Übersetzung nicht ausreichend)
- auf eigene Beispiele übertragen, Thema daran veranschaulichen
- Quellen angemessen zitieren (konsistent!)

Anforderungen Ausarbeitung

Format & Struktur

- 5–7 A4-Seiten, PDF-Format, LaTeX verwenden
- sinnvoll strukturieren (Titel, Einleitung, Grundlagen, Hauptteil, Schluss)

Inhalt

- Definitionen und Kernaussagen herausfiltern
- in eigenen Worten wiedergeben (reine Übersetzung nicht ausreichend)
- auf eigene Beispiele übertragen, Thema daran veranschaulichen
- Quellen angemessen zitieren (konsistent!)

Buch:	Autor, Titel, Verlag, Erscheinungsjahr, ggf. Auflage
Buchbeitrag:	Autor, Titel, Buchtitel, Seiten, Erscheinungsjahr, ggf. Herausgeber, Verlag
Fachzeitschrift:	Autor, Titel, Zeitschrift, Band und Nummer, Seiten, Erscheinungsjahr
(~~Wikipedia~~)	*ok zum Nachgucken oder für Bilderquellen*

Bewertungskriterien

1. Lesbarkeit
(Struktur, Länge, Sprache, Verständlichkeit, Verknüpfungen)

Bewertungskriterien

1. Lesbarkeit
   (Struktur, Länge, Sprache, Verständlichkeit, Verknüpfungen)
2. technische Qualität
   (Definitionen, Anschaulichkeit, Korrektheit, Genauigkeit)

Bewertungskriterien

1. Lesbarkeit
   (Struktur, Länge, Sprache, Verständlichkeit, Verknüpfungen)
2. technische Qualität
   (Definitionen, Anschaulichkeit, Korrektheit, Genauigkeit)
3. Referenzen und Eigenanteil
   (Quellenangaben, Literaturverzeichnis, eigene Formulierungen)

technische universität
dortmund

Effiziente Algorithmen und
Komplexitätstheorie /
Informatik

Aufteilungs- und Zuweisungsalgorithmen

Proseminar
im Wintersemester 2016/2017

Anja Rey, 16. November 2016

Zeitplan

Beginn

- Mi 19.10.16 Einleitung, Themen
- Do 20.10.16 Beginn Präsentationskurs

Themenvergabe

- Thema auswählen bis 25.10 unter

  `http://doodle.com/poll/37hfa3gxyn4xdc7n`

Ausarbeitungsphase

- Mi 26.10.16 Besprechung
- Mi 02.11.16 & Mi 09.11.16 Fragen & Bearbeitung
- Di 15.11.16 erste Abgabe Ausarbeitung

— Rechtzeitig Vortragsplanung beginnen! —

Reviewphase

- Verteilung Reviews (per E-Mail)
- Mi 16.11.16 Besprechung
- Mi 23.11.16 Fragen & Bearbeitung
- Do 24.11.16 Abgabe Reviews
- Mi 30.11.16 Fragen & Bearbeitung
- Mi 30.11.16 zweite Abgabe Ausarbeitung

Zwischenbesprechung

- 8. Woche Einzeltermine
- Do 08.12.16 Ende Präsentationskurs

Vortragsphase und Ende

- 9.–15. Woche: Vorträge
- Feedback am Ende

Reviewform

Thema zusammenfassen

- Worum geht es in der Ausarbeitung?
- zwei bis drei Sätze

Reviewform

Thema zusammenfassen

- Worum geht es in der Ausarbeitung?
- zwei bis drei Sätze

Bewertung

- zu jedem Kriterium 1–5 Punkte vergeben

5	hervorragend, (fast) nichts auszusetzen
4	gut, fast alles erfüllt, an wenigen Stellen verbesserungswürdig
3	mittelmäßig, Kriterien halbwegs erfüllt, ein paar Mängel
2	verbesserungswürdig, wenige gute Aspekte
1	reicht nicht aus, viele Fehler

Reviewform

Thema zusammenfassen

- Worum geht es in der Ausarbeitung?
- zwei bis drei Sätze

Bewertung

- zu jedem Kriterium 1–5 Punkte vergeben

5	hervorragend, (fast) nichts auszusetzen
4	gut, fast alles erfüllt, an wenigen Stellen verbesserungswürdig
3	mittelmäßig, Kriterien halbwegs erfüllt, ein paar Mängel
2	verbesserungswürdig, wenige gute Aspekte
1	reicht nicht aus, viele Fehler

- zu jedem Kriterium eine kurze Begründung forumulieren

Reviewform

Thema zusammenfassen

- Worum geht es in der Ausarbeitung?
- zwei bis drei Sätze

Bewertung

- zu jedem Kriterium 1–5 Punkte vergeben
  - 5 hervorragend, (fast) nichts auszusetzen
  - 4 gut, fast alles erfüllt, an wenigen Stellen verbesserungswürdig
  - 3 mittelmäßig, Kriterien halbwegs erfüllt, ein paar Mängel
  - 2 verbesserungswürdig, wenige gute Aspekte
  - 1 reicht nicht aus, viele Fehler
- zu jedem Kriterium eine kurze Begründung forumulieren
- *Minor Issues:* in Stichpunkten ausgewählte Details festhalten (positive Details, Kritikpunkte, Fehlerdetails)

Reviewform

Bewertungskriterien

1. Lesbarkeit
   (Struktur, Verknüpfungen, Länge, Sprache)

Reviewform

Bewertungskriterien

1. Lesbarkeit
   (Struktur, Verknüpfungen, Länge, Sprache)
2. technische Qualität
   (Definitionen, Anschaulichkeit, Korrektheit)

Reviewform

Bewertungskriterien

1. Lesbarkeit
   (Struktur, Verknüpfungen, Länge, Sprache)
2. technische Qualität
   (Definitionen, Anschaulichkeit, Korrektheit)
3. Referenzen und Eigenanteil
   (Quellenangaben, Literaturverzeichnis, Eigenanteil)

Reviewform

Bewertungskriterien

1. Lesbarkeit
   (Struktur, Verknüpfungen, Länge, Sprache)
2. technische Qualität
   (Definitionen, Anschaulichkeit, Korrektheit)
3. Referenzen und Eigenanteil
   (Quellenangaben, Literaturverzeichnis, Eigenanteil)
4. Gesamtbewertung

Reviewform

Bewertungskriterien

1. Lesbarkeit
   (Struktur, Verknüpfungen, Länge, Sprache)
2. technische Qualität
   (Definitionen, Anschaulichkeit, Korrektheit)
3. Referenzen und Eigenanteil
   (Quellenangaben, Literaturverzeichnis, Eigenanteil)
4. Gesamtbewertung

weitere Aspekte

- *Confidence:* Erfahrung mit dem Thema einschätzen

Reviewform

Bewertungskriterien

1. Lesbarkeit
   (Struktur, Verknüpfungen, Länge, Sprache)
2. technische Qualität
   (Definitionen, Anschaulichkeit, Korrektheit)
3. Referenzen und Eigenanteil
   (Quellenangaben, Literaturverzeichnis, Eigenanteil)
4. Gesamtbewertung

weitere Aspekte

- *Confidence:* Erfahrung mit dem Thema einschätzen
- Zusätzliche Angaben, die nicht weitergegeben werden

Reviewform

WICHTIG

- Feedback-Regeln einhalten: sachliche Bewertung aus eigener Perspektive, positive und konstruktive Kritik

Reviewform

WICHTIG

- Feedback-Regeln einhalten: sachliche Bewertung aus eigener Perspektive, positive und konstruktive Kritik
- Beurteilung hat keine negative Auswirkung auf Bewertung
- Rückmeldung nutzen zur Überarbeitung!

Reviewform

WICHTIG

- Feedback-Regeln einhalten: sachliche Bewertung aus eigener Perspektive, positive und konstruktive Kritik
- Beurteilung hat keine negative Auswirkung auf Bewertung
- Rückmeldung nutzen zur Überarbeitung!
- *single blind* (Reviewer nicht bekannt)

Reviewform

WICHTIG

- Feedback-Regeln einhalten: sachliche Bewertung aus eigener Perspektive, positive und konstruktive Kritik
- Beurteilung hat keine negative Auswirkung auf Bewertung
- Rückmeldung nutzen zur Überarbeitung!
- *single blind* (Reviewer nicht bekannt)
- jeder erhält sein Review und ein separates Feedback im Zwischengespräch

Ablauf der Vorträge

Vortrag

- 30 Minuten Präsentation des Themas

Ablauf der Vorträge

Vortrag

- 30 Minuten Präsentation des Themas
- Folien per E-Mail schicken, auf USB-Stick mitbringen oder eigenes Notebook (vorher testen!)

Ablauf der Vorträge

Vortrag

- 30 Minuten Präsentation des Themas
- Folien per E-Mail schicken, auf USB-Stick mitbringen oder eigenes Notebook (vorher testen!)
- weitere benötigte Materialien (Whiteboardmarker, Kopien o. Ä.) rechtzeitig ankündigen

Ablauf der Vorträge

Vortrag

- 30 Minuten Präsentation des Themas
- Folien per E-Mail schicken, auf USB-Stick mitbringen oder eigenes Notebook (vorher testen!)
- weitere benötigte Materialien (Whiteboardmarker, Kopien o. Ä.) rechtzeitig ankündigen

Fragen & Diskussion

- bis zu 15 Minuten Zeit für Fragen

Ablauf der Vorträge

Vortrag

- 30 Minuten Präsentation des Themas
- Folien per E-Mail schicken, auf USB-Stick mitbringen oder eigenes Notebook (vorher testen!)
- weitere benötigte Materialien (Whiteboardmarker, Kopien o. Ä.) rechtzeitig ankündigen

Fragen & Diskussion

- bis zu 15 Minuten Zeit für Fragen
- jeder stellt eine Frage zu mindestens einem Vortrag

Ablauf der Vorträge

Vortrag

- 30 Minuten Präsentation des Themas
- Folien per E-Mail schicken, auf USB-Stick mitbringen oder eigenes Notebook (vorher testen!)
- weitere benötigte Materialien (Whiteboardmarker, Kopien o. Ä.) rechtzeitig ankündigen

Fragen & Diskussion

- bis zu 15 Minuten Zeit für Fragen
- jeder stellt eine Frage zu mindestens einem Vortrag
- Ausarbeitungen werden zur Verfügung gestellt (Passwort)
  – bitte Bescheid geben, falls nicht einverstanden