Algorithmen zur präferenzbasierten Entscheidungsfindung

Proseminar im Wintersemester 2018/2019

Anja Rey

Effiziente Algorithmen und Komplexitätstheorie, Informatik

12. Oktober 2018

technische universität
dortmund

Organisatorisches

Proseminar (3 CP)

Präsentationskurs (1 CP)

Organisatorisches

Proseminar (3 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
  - Vorversion zur Übersicht in Form eines *Abstracts*

Präsentationskurs (1 CP)

Organisatorisches

Proseminar (3 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
  - Vorversion zur Übersicht in Form eines *Abstracts*
- *Peer-Review* zu einer anderen Ausarbeitung

Präsentationskurs (1 CP)

Organisatorisches

Proseminar (3 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
  - Vorversion zur Übersicht in Form eines *Abstracts*
- *Peer-Review* zu einer anderen Ausarbeitung
- erfolgreich absolvierte Zwischengespräche
  1. Rückmeldung der Seminarleiterin zum Abstract
  2. Rückmeldung der Seminarleiterin zur Ausarbeitung, Vorstellung eines schlüssigen Vortragskonzepts

Präsentationskurs (1 CP)

Organisatorisches

Proseminar (3 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
  - Vorversion zur Übersicht in Form eines *Abstracts*
- *Peer-Review* zu einer anderen Ausarbeitung
- erfolgreich absolvierte Zwischengespräche
  1. Rückmeldung der Seminarleiterin zum Abstract
  2. Rückmeldung der Seminarleiterin zur Ausarbeitung, Vorstellung eines schlüssigen Vortragskonzepts
- 30-minütiger Vortrag des Themas (exkl. Fragen & Diskussion)

Präsentationskurs (1 CP)

Organisatorisches

Proseminar (3 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
  - Vorversion zur Übersicht in Form eines *Abstracts*
- *Peer-Review* zu einer anderen Ausarbeitung
- erfolgreich absolvierte Zwischengespräche
  1. Rückmeldung der Seminarleiterin zum Abstract
  2. Rückmeldung der Seminarleiterin zur Ausarbeitung, Vorstellung eines schlüssigen Vortragskonzepts
- 30-minütiger Vortrag des Themas (exkl. Fragen & Diskussion)
- Frage zu mindestens einem Vortrag

Präsentationskurs (1 CP)

Organisatorisches

Proseminar (3 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
  - Vorversion zur Übersicht in Form eines *Abstracts*
- *Peer-Review* zu einer anderen Ausarbeitung
- erfolgreich absolvierte Zwischengespräche
  1. Rückmeldung der Seminarleiterin zum Abstract
  2. Rückmeldung der Seminarleiterin zur Ausarbeitung, Vorstellung eines schlüssigen Vortragskonzepts
- 30-minütiger Vortrag des Themas (exkl. Fragen & Diskussion)
- Frage zu mindestens einem Vortrag

Präsentationskurs (1 CP)

begleitend: 4 Blöcke, je Fr 12–16 Uhr in OH 14, 304

Zeitplan

Beginn	
Fr 12.10.18	Einleitung, Themenvorstellung

Zeitplan

Beginn

Fr 12.10.18 Einleitung, Themenvorstellung

Themenvergabe

Literaturrecherche

Thema auswählen bis 18.10 im moodle

Zeitplan

Beginn

Fr 12.10.18 Einleitung, Themenvorstellung

Themenvergabe

Literaturrecherche

Thema auswählen bis 18.10 im moodle

Ausarbeitungsphase

Fr 19.10.18 / 26.10.18 Präsentationskurs 1

Fr 02.11.18 Besprechung & Bearbeitung

Do 08.11.18 erste Abgabe Ausarbeitung und Abstract

– Frühzeitig inhaltliche Fragen klären! –

– Rechtzeitig Vortragsplanung beginnen! –

Zeitplan

Beginn

Fr 12.10.18 Einleitung, Themenvorstellung

Themenvergabe

Literaturrecherche

Thema auswählen bis 18.10 im moodle

Ausarbeitungsphase

Fr 19.10.18 / 26.10.18 Präsentationskurs 1

Fr 02.11.18 Besprechung & Bearbeitung

Do 08.11.18 erste Abgabe Ausarbeitung und Abstract

– Frühzeitig inhaltliche Fragen klären! –

– Rechtzeitig Vortragsplanung beginnen! –

Reviewphase

Verteilung Reviews (per E-Mail)

Fr 09.11.18 / 16.11.18 Präsentationskurs 2

6. & 7. Woche Zwischenbesprechung 1

Do 22.11.18 Abgabe Reviews

Fr 23.11.18 / 30.11.18 Präsentationskurs 3

Di 04.12.18 zweite Abgabe Ausarbeitung

Zeitplan

Beginn

Fr 12.10.18 Einleitung, Themenvorstellung

Themenvergabe

Literaturrecherche

Thema auswählen bis 18.10 im moodle

Ausarbeitungsphase

Fr 19.10.18 / 26.10.18 Präsentationskurs 1

Fr 02.11.18 Besprechung & Bearbeitung

Do 08.11.18 erste Abgabe Ausarbeitung und Abstract

– Frühzeitig inhaltliche Fragen klären! –

– Rechtzeitig Vortragsplanung beginnen! –

Reviewphase

Verteilung Reviews (per E-Mail)

Fr 09.11.18 / 16.11.18 Präsentationskurs 2

6. & 7. Woche Zwischenbesprechung 1

Do 22.11.18 Abgabe Reviews

Fr 23.11.18 / 30.11.18 Präsentationskurs 3

Di 04.12.18 zweite Abgabe Ausarbeitung

Vortragsvorbereitung

Fr 07.12.18 / 14.12.18 Präsentationskurs 4

10. & 11. Woche Zwischenbesprechung 2

Fr 21.12.18 offenes Arbeiten

Zeitplan

Beginn

Fr 12.10.18 Einleitung, Themenvorstellung

Themenvergabe

Literaturrecherche

Thema auswählen bis 18.10 im moodle

Ausarbeitungsphase

Fr 19.10.18 / 26.10.18 Präsentationskurs 1

Fr 02.11.18 Besprechung & Bearbeitung

Do 08.11.18 erste Abgabe Ausarbeitung und Abstract

– Frühzeitig inhaltliche Fragen klären! –

– Rechtzeitig Vortragsplanung beginnen! –

Reviewphase

Verteilung Reviews (per E-Mail)

Fr 09.11.18 / 16.11.18 Präsentationskurs 2

6. & 7. Woche Zwischenbesprechung 1

Do 22.11.18 Abgabe Reviews

Fr 23.11.18 / 30.11.18 Präsentationskurs 3

Di 04.12.18 zweite Abgabe Ausarbeitung

Vortragsvorbereitung

Fr 07.12.18 / 14.12.18 Präsentationskurs 4

10. & 11. Woche Zwischenbesprechung 2

Fr 21.12.18 offenes Arbeiten

Vortragsphase und Ende

Fr 11., 18., 25.01. & 01.02.2019 Vorträge

Abschluss-Feedback

Fragen?

Adressen

- Informationen unter
  `http://ls2-www.cs.uni-dortmund.de/~rey/wise1819proseminar`
- Themen und weitere Materialien unter
  `https://moodle.tu-dortmund.de/course/view.php?id=12757`
- Abgaben und weitere Fragen an anja.rey@tu-dortmund.de
- Zwischengespräche und weitere Fragen in OH 14, 313
- weitere Fragen in Seminarterminen und Sprechstunden

  Mo 13:30 Uhr bis 14:30 Uhr

Bewertung

Benotung auf Basis der Ausarbeitung und des Vortrags

Bewertung

Benotung auf Basis der Ausarbeitung und des Vortrags

Bewertungskriterien der Ausarbeitung

1. Lesbarkeit
   (Struktur, Anteile, Verknüpfungen, Fachsprache, Verständlichkeit, Breite)

Bewertung

Benotung auf Basis der Ausarbeitung und des Vortrags

Bewertungskriterien der Ausarbeitung

1. Lesbarkeit
   (Struktur, Anteile, Verknüpfungen, Fachsprache, Verständlichkeit, Breite)
2. Technische Qualität
   (formale Definitionen und Aussagen, Anschaulichkeit, Korrektheit, Vollständigkeit, Genauigkeit, Tiefe)

Bewertung

Benotung auf Basis der Ausarbeitung und des Vortrags

Bewertungskriterien der Ausarbeitung

1. Lesbarkeit
   (Struktur, Anteile, Verknüpfungen, Fachsprache, Verständlichkeit, Breite)
2. Technische Qualität
   (formale Definitionen und Aussagen, Anschaulichkeit, Korrektheit, Vollständigkeit, Genauigkeit, Tiefe)
3. Referenzen und Eigenanteil
   (Quellenangaben, Literaturverzeichnis, Auswahl von Wesentlichem, eigene Formulierungen und Ausführungen)

Themenübersicht

Algorithmen zur präferenzbasierten Entscheidungsfindung

- Wahlverfahren
- faire Aufteilung teilbarer und unteilbarer Güter
- *Matching*-Algorithmen
- Koalitionsbildung

Literatur

- F. Brandt, V. Conitzer, U. Endriss, J. Lang und A. Procaccia, Editoren.
  ***Handbook of Computational Social Choice*.**
  Cambridge University Press, 2016.
- U. Endriss, Editor.
  ***Trends in Computational Social Choice*.**
  AI Access, 2017.
- D. Manlove
  ***Algorithmics Of Matching Under Preferences*.**
  Volume 2 of Theoretical computer science. World Scientific Publishing, 2013.

. . . und jede Menge Referenzen darin.

Wahlverfahren

Wahl:

- Kandidaten: $C = \{$  $\}$
- Wähler: V

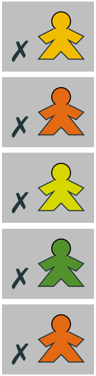
Wahlsystem: $(C, V) \mapsto$ **Gewinner**

Wahlverfahren

Wahl:

- Kandidaten: $C = \{$  $\}$
- Wähler: V: Auswahl

Wahlsystem: $(C, V) \mapsto$ **Gewinner**

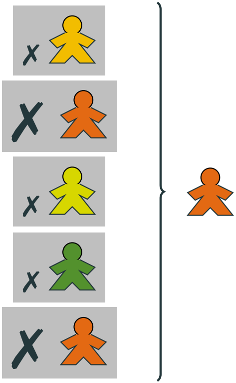


Wahlverfahren

Wahl:

- Kandidaten: $C = \{$  $\}$
- Wähler: V: Auswahl

Wahlsystem: $(C, V) \mapsto$ **Gewinner**

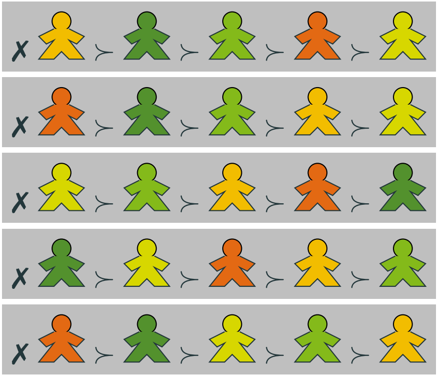


Wahlverfahren

Wahl:

- Kandidaten: $C =$ 
- Wähler: V Auswahl, Präferenzlisten, ...

Wahlsystem: $(C, V) \mapsto$ **Gewinner**

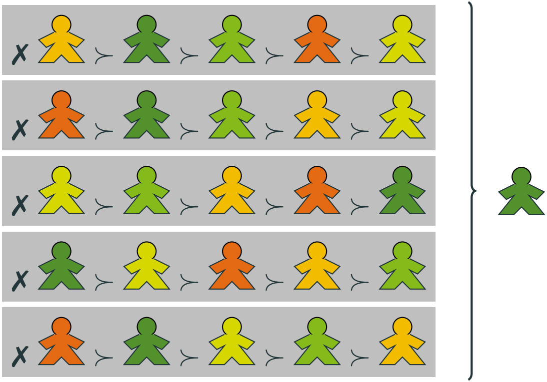


Wahlverfahren

Wahl:

- Kandidaten: $C =$ 
- Wähler: V Auswahl, Präferenzlisten, ...

Wahlsystem: $(C, V) \mapsto$ **Gewinner**



Wahlverfahren

Wahl:

- Kandidaten: $C = \{$  $\}$
- Wähler: V Auswahl, Präferenzlisten, ...

Wahlsystem: $(C, V) \mapsto$ **Gewinner**

- *Scoring*-Regeln: Pluralität, Borda-Regel, ...
- vergleichsbasiert, rundenbasiert, ...

Wahlverfahren

Wahl:

- Kandidaten: $C = \{$  $\}$
- Wähler: V Auswahl, Präferenzlisten, ...

Wahlsystem: $(C, V) \mapsto$ **Gewinner**

- *Scoring*-Regeln: Pluralität, Borda-Regel, ...
- vergleichsbasiert, rundenbasiert, ...

Fragestellungen:

- *Wer gewinnt?*
- *Ist $c \in C$ ein (eindeutiger/nicht-eindeutiger) Gewinner?*

Wahlverfahren: Eigenschaften

Was ist ein gutes & faires Wahlverfahren?

- grundlegende Kriterien: Anonymität, Neutralität, Souveränität, ...

W. Zwicker. **Introduction to the Theory of Voting.** In Brand et al.: Handbook of Computational Social Choice, Kapitel 2. Cambridge University Press, 2016.

F. Brandt, M. Brill und P. Harrenstein. **Tournament Solutions.** In Brand et al.: Handbook of Computational Social Choice, Kapitel 3. Cambridge University Press, 2016.

R. Meir. **Iterative Voting.** In U. Endriss: Trends in Computational Social Choice, Kapitel 4. AI Access, 2017.

Wahlverfahren: Eigenschaften

Was ist ein gutes & faires Wahlverfahren?

- grundlegende Kriterien: Anonymität, Neutralität, Souveränität, ...
- strukturelle Konsistenz: Condorcet-Eigenschaft, Monotonie, *Responsiveness*, Unabhängigkeit von irrelevanten Alternativen, Pareto-Kriterium, ...

W. Zwicker. **Introduction to the Theory of Voting.** In Brand et al.: Handbook of Computational Social Choice, Kapitel 2. Cambridge University Press, 2016.

F. Brandt, M. Brill und P. Harrenstein. **Tournament Solutions.** In Brand et al.: Handbook of Computational Social Choice, Kapitel 3. Cambridge University Press, 2016.

R. Meir. **Iterative Voting.** In U. Endriss: Trends in Computational Social Choice, Kapitel 4. AI Access, 2017.

Wahlverfahren: Eigenschaften

Was ist ein gutes & faires Wahlverfahren?

- grundlegende Kriterien: Anonymität, Neutralität, Souveränität, ...
- strukturelle Konsistenz: Condorcet-Eigenschaft, Monotonie, *Responsiveness*, Unabhängigkeit von irrelevanten Alternativen, Pareto-Kriterium, ... $\rightsquigarrow$ Unmöglichkeitstheoreme

W. Zwicker. **Introduction to the Theory of Voting.** In Brand et al.: Handbook of Computational Social Choice, Kapitel 2. Cambridge University Press, 2016.

F. Brandt, M. Brill und P. Harrenstein. **Tournament Solutions.** In Brand et al.: Handbook of Computational Social Choice, Kapitel 3. Cambridge University Press, 2016.

R. Meir. **Iterative Voting.** In U. Endriss: Trends in Computational Social Choice, Kapitel 4. AI Access, 2017.

Wahlverfahren: Eigenschaften

Was ist ein gutes & faires Wahlverfahren?

- grundlegende Kriterien: Anonymität, Neutralität, Souveränität, ...
- strukturelle Konsistenz: Condorcet-Eigenschaft, Monotonie, *Responsiveness*, Unabhängigkeit von irrelevanten Alternativen, Pareto-Kriterium, ... $\rightsquigarrow$ Unmöglichkeitstheoreme
- effiziente Gewinnerbestimmung
- schwer zu beeinflussen (Manipulation, Kontrolle, Bestechung)

W. Zwicker. **Introduction to the Theory of Voting.** In Brand et al.: Handbook of Computational Social Choice, Kapitel 2. Cambridge University Press, 2016.

F. Brandt, M. Brill und P. Harrenstein. **Tournament Solutions.** In Brand et al.: Handbook of Computational Social Choice, Kapitel 3. Cambridge University Press, 2016.

R. Meir. **Iterative Voting.** In U. Endriss: Trends in Computational Social Choice, Kapitel 4. AI Access, 2017.

Wahlverfahren: Eigenschaften

Was ist ein gutes & faires Wahlverfahren?

- grundlegende Kriterien: Anonymität, Neutralität, Souveränität, ...
- strukturelle Konsistenz: Condorcet-Eigenschaft, Monotonie, *Responsiveness*, Unabhängigkeit von irrelevanten Alternativen, Pareto-Kriterium, ... $\rightsquigarrow$ Unmöglichkeitstheoreme
- effiziente Gewinnerbestimmung
- schwer zu beeinflussen (Manipulation, Kontrolle, Bestechung)

Weitere Verfahren: *Tournament solutions*, iteratives Wählen

W. Zwicker. **Introduction to the Theory of Voting.** In Brand et al.: Handbook of Computational Social Choice, Kapitel 2. Cambridge University Press, 2016.

F. Brandt, M. Brill und P. Harrenstein. **Tournament Solutions.** In Brand et al.: Handbook of Computational Social Choice, Kapitel 3. Cambridge University Press, 2016.

R. Meir. **Iterative Voting.** In U. Endriss: Trends in Computational Social Choice, Kapitel 4. AI Access, 2017.

Wahlverfahren: Eigenschaften

Was ist ein gutes & faires Wahlverfahren?

- grundlegende Kriterien: Anonymität, Neutralität, Souveränität, ...
- strukturelle Konsistenz: Condorcet-Eigenschaft, Monotonie, *Responsiveness*, Unabhängigkeit von irrelevanten Alternativen, Pareto-Kriterium, ... $\rightsquigarrow$ Unmöglichkeitstheoreme Thema 1
- effiziente Gewinnerbestimmung
- schwer zu beeinflussen (Manipulatio Thema 2 lle, Bestechu Thema 3

Weitere Verfahren: *Tournament solutions*, iteratives Wählen

W. Zwicker. **Introduction to the Theory of Voting.** In Brand et al.: Handbook of Computational Social Choice, Kapitel 2. Cambridge University Press, 2016.

F. Brandt, M. Brill und P. Harrenstein. **Tournament Solutions.** In Brand et al.: Handbook of Computational Social Choice, Kapitel 3. Cambridge University Press, 2016.

R. Meir. **Iterative Voting.** In U. Endriss: Trends in Computational Social Choice, Kapitel 4. AI Access, 2017.

Judgment Aggregation

Ziel: Aggregation von logischen Argumenten

Judgment Aggregation

Ziel: Aggregation von logischen Argumenten

Beispel: Diskursives Dilemma

- Urteil über Vertragsbruch
- Rechtslage: Vertragsbruch $\iff$ Vertrag $\wedge$ Bruch

Judgment Aggregation

Ziel: Aggregation von logischen Argumenten

Beispel: Diskursives Dilemma

- Urteil über Vertragsbruch
- Rechtslage: Vertragsbruch $\iff$ Vertrag $\wedge$ Bruch
- 3 Einzelurteile:

Richter	Vertrag	Bruch	Vertragsbruch
	1	1	1
	1	0	0
	0	1	0

Judgment Aggregation

Ziel: Aggregation von logischen Argumenten

Beispel: Diskursives Dilemma

- Urteil über Vertragsbruch
- Rechtslage: Vertragsbruch $\iff$ Vertrag $\wedge$ Bruch
- 3 Einzelurteile:

Richter	Vertrag	Bruch	Vertragsbruch
	1	1	1
	1	0	0
	0	1	0
Mehrheit	1	1	0

Judgment Aggregation

Ziel: Aggregation von logischen Argumenten

Beispel: Diskursives Dilemma

- Urteil über Vertragsbruch
- Rechtslage: Vertragsbruch $\iff$ Vertrag $\land$ Bruch
- 3 Einzelurteile:

Richter	Vertrag	Bruch	Vertragsbruch
	1	1	1
	1	0	0
	0	1	0
Mehrheit	1	1	0

Verfahren: Quotenregeln, distanzbasierte Regeln

Konzepte: Agenda-Eigenschaften, z. B. Vollständigkeit, Konsistenz, ...

U. Endriss **Judgment Aggregation.** In Brand et al.: Handbook of Computational Social Choice, Kapitel 17. Cambridge University Press, 2016.

Judgment Aggregation

Ziel: Aggregation von logischen Argumenten

Beispel: Diskursives Dilemma

- Urteil über Vertragsbruch
- Rechtslage: Vertragsbruch $\iff$ Vertrag $\land$ Bruch
- 3 Einzelurteile:

Richter	Vertrag	Bruch	Vertragsbruch
	1	1	1
	1	0	0
	0	1	0
Mehrheit	1	1	0

Thema 4

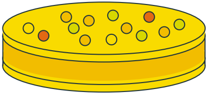
Verfahren: Quotenregeln, distanzbasierte Regeln

Thema 5

Konzepte: Agenda-Eigenschaften, z. B. …digkeit, Konsistenz, ...

U. Endriss **Judgment Aggregation.** In Brand et al.: Handbook of Computational Social Choice, Kapitel 17. Cambridge University Press, 2016.

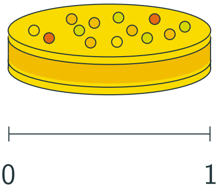
Aufteilung teilbarer Güter



Gut: inhomogener *Kuchen*

Ziel: jeder bekommt eine *faire* Portion

Aufteilung teilbarer Güter



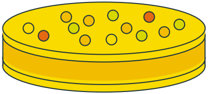
Gut: inhomogener *Kuchen*

Ziel: jeder bekommt eine *faire* Portion

Definition

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

Aufteilung teilbarer Güter



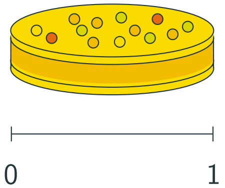
Gut: inhomogener *Kuchen*

Ziel: jeder bekommt eine *faire* Portion

Definition

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

Aufteilung teilbarer Güter



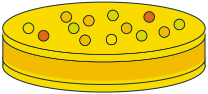
Gut: inhomogener *Kuchen*

Ziel: jeder bekommt eine *faire* Portion

Definition

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



Gut: inhomogener *Kuchen*

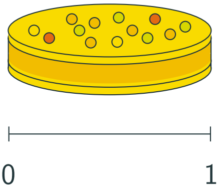
Ziel: jeder bekommt eine *faire* Portion

Definition

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



Gut: inhomogener *Kuchen*

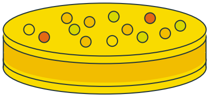
Ziel: jeder bekommt eine *faire* Portion

Definition

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

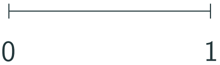
- $v_i([a,a]) = 0 \; \forall a \in [0,1]$ (Nicht-Atomarität);
- $v_i([0,1]) = 1$ (Normalisierung);
- für jedes Teilintervall $X \subseteq [0,1]$: $v_i(X) \geq 0$ (Nicht-Negativität);
- für jedes Teilintervall $[a,b] \subseteq [0,1]$ und λ: $0 \leq \lambda \leq 1$, existiert ein Punkt $c \in [a,b]$ mit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

Aufteilung teilbarer Güter



Gut: inhomogener *Kuchen*

Ziel: jeder bekommt eine *faire* Portion



Definition

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

A. Procaccia. **Cake Cutting Algorithms.** In Brand et al.: Handbook of Computational Social Choice, Kapitel 13. Cambridge University Press, 2016.

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

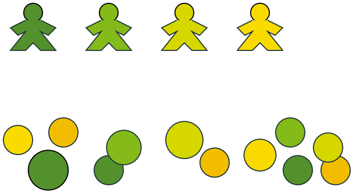
Thema 6

Thema 7

A. Procaccia. **Cake Cutting Algorithms.** In Brand et al.: Handbook of Computational Social Choice, Kapitel 13. Cambridge University Press, 2016.

Aufteilung unteilbarer Güter

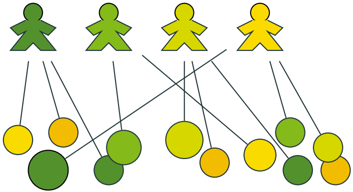
Güter: Objekte/Ressourcen mit unterschiedlichem Wert für Agenten



Aufteilung unteilbarer Güter

Güter: Objekte/Ressourcen mit unterschiedlichem Wert für Agenten

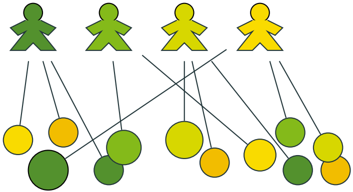
Ziel: jeder bekommt einen *optimalen* Anteil (1 : *many*)



Aufteilung unteilbarer Güter

Güter: Objekte/Ressourcen mit unterschiedlichem Wert für Agenten

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

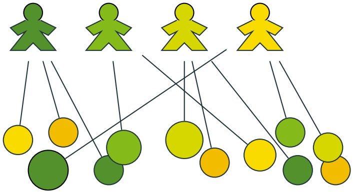


Definition (*MultiAgent Resource Allocation* (MARA))

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



Güter: Objekte/Ressourcen mit unterschiedlichem Wert für Agenten

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

Definition (*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.

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

Protokolle:

- *Adjusted-Winner*-Verfahren
- Proportionales Aufteilungsverfahren
- Approximations-Algorithmen

S. Bouveret, Y. Chevaleyre und N. Maudet. **Fair Allocation of Indivisible Goods.** In Brand et al.: Handbook of Computational Social Choice, Kapitel 12. Cambridge University Press, 2016.

E. Markakis. **Approximation Algorithms and Hardness Results for Fair Division.** In U. Endriss: Trends in Computational Social Choice, Kapitel 12. AI Access, 2017.

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

Protokolle:

- *Adjusted-Winner*-Verfahren
- Proportionales Aufteilungsve
- Approximations-Algorithmen

Thema 8

Thema 9

S. Bouveret, Y. Chevaleyre und N. Maudet. **Fair Allocation of Indivisible Goods.** In Brand et al.: Handbook of Computational Social Choice, Kapitel 12. Cambridge University Press, 2016.

E. Markakis. **Approximation Algorithms and Hardness Results for Fair Division.** In U. Endriss: Trends in Computational Social Choice, Kapitel 12. AI Access, 2017.

Matching-Algorithmen

Instanz:

- Agenten
- ordinale Präferenzen über Teilmengen der Agenten
- andere Bedingungen: Kapazitäten etc.

Matching-Algorithmen

Instanz:

- Agenten
- ordinale Präferenzen über Teilmengen der Agenten
- andere Bedingungen: Kapazitäten etc.

Ziel: faires oder optimales (*Matching*) finden

Matching-Algorithmen

Instanz:

- Agenten
- ordinale Präferenzen über Teilmengen der Agenten
- andere Bedingungen: Kapazitäten etc.

Ziel: faires oder optimales (*Matching*) finden

Unterscheidung zwischen Agenten mit

- bipartiten und einseitigen Präferenzen
- bipartiten und beidseitigen Präferenzen
- nicht-bipartiten Präferenzen

Matching-Algorithmen

Instanz:

- Agenten
- ordinale Präferenzen über Teilmengen der Agenten
- andere Bedingungen: Kapazitäten etc.

Ziel: faires oder optimales (*Matching*) finden

Unterscheidung zwischen Agenten mit

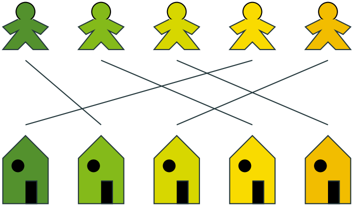
- bipartiten und einseitigen Präferenzen
- bipartiten und beidseitigen Präferenzen
- nicht-bipartiten Präferenzen

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

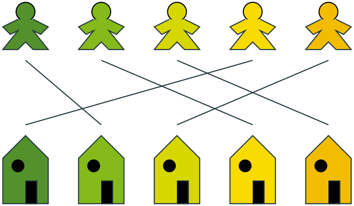
Das *House-Allocation*-Problem

Präferenzen: einseitig von Bewerbern über Häuser

Ziel: (1 : 1)-Zuweisung



Das *House-Allocation*-Problem



Präferenzen: einseitig von Bewerbern über Häuser

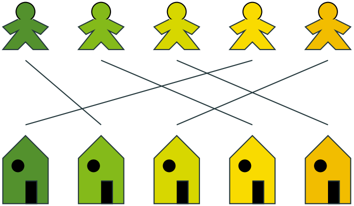
Ziel: (1 : 1)-Zuweisung

Optimalitätskriterien

- Pareto-Optimalität
- Popularität
- profilbasierte Optimalität: rang-maximal, gierig & großzügig

D. Manlove. **Algorithmics Of Matching Under Preferences.** Kapitel 6, 7 und 8. Cambridge University Press, 2016.

Das *House-Allocation*-Problem



Präferenzen: einseitig von Bewerbern über Häuser

Ziel: (1 : 1)-Zuweisung

Optimalitätskriterien

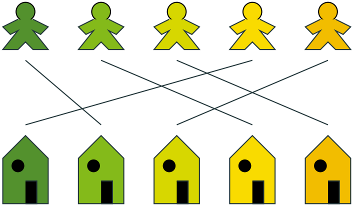
- Pareto-Optimalität
- Popularität
- profilbasierte Optimalität: rang-maximal, gierig & großzügig

Varianten:

- *Housing Markets*: bestehendes Matching, individuelle Rationalität
- gewichtete Präferenzen
- (*many* : 1)-Verallgemeinerung: WG mit Kapazitäten

D. Manlove. **Algorithmics Of Matching Under Preferences.** Kapitel 6, 7 und 8. Cambridge University Press, 2016.

Das *House-Allocation*-Problem



Präferenzen: einseitig von Bewerbern über Häuser

Ziel: (1 : 1)-Zuweisung

Optimalitätskriterien

- Pareto-Optimalität — Thema 10
- Popularität — Thema 11
- profilbasierte Optimalität: rang-maximal, gierig & großzügig — Thema 12

Varianten:

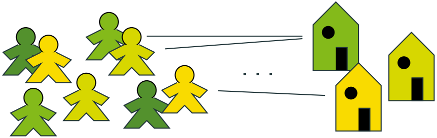
- *Housing Markets*: bestehendes Matching, individuelle Rationalität
- gewichtete Präferenzen
- (*many* : 1)-Verallgemeinerung: WG mit Kapazitäten

D. Manlove. **Algorithmics Of Matching Under Preferences.** Kapitel 6, 7 und 8. Cambridge University Press, 2016.

Zuweisung von Ärzten auf Krankenhäuser

Präferenzen: beidseitig

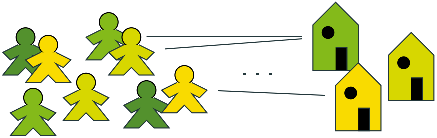
Ziel: ($many$: 1)-Zuweisung



Zuweisung von Ärzten auf Krankenhäuser

Präferenzen: beidseitig

Ziel: (*many* : 1)-Zuweisung



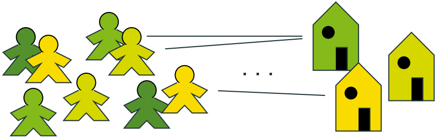
Definition (Das *Hospital-Residents*-Problem (HR))

Eine HR-Instanz besteht aus

Zuweisung von Ärzten auf Krankenhäuser

Präferenzen: beidseitig

Ziel: $(\textit{many} : 1)$-Zuweisung



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

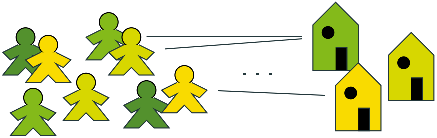
Eine HR-Instanz besteht aus

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

Zuweisung von Ärzten auf Krankenhäuser

Präferenzen: beidseitig

Ziel: (*many* : 1)-Zuweisung



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

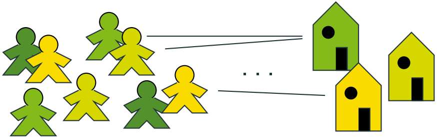
Eine HR-Instanz besteht aus

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

Zuweisung von Ärzten auf Krankenhäuser

Präferenzen: beidseitig

Ziel: $(\mathit{many} : 1)$-Zuweisung



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

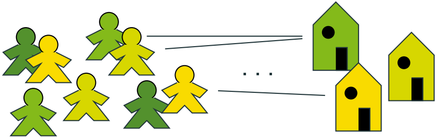
Eine HR-Instanz besteht aus

- disjunkten Agentenmengen $R = \{r_1, \ldots, r_{n_1}\}$ und $H = \{h_1, \ldots, h_{n_2}\}$,
- Kapazität c_j für jedes $h_j \in H$,
- Menge $E \subseteq R \times H$ der akzeptablen Paare, $m = \|E\|$, mit
  - $A(r_i) = \{h_j \in H \mid (r_i, h_j) \in E\}$ akzeptable Krankenhäuser für $r_i \in R$
  - $A(h_j) = \{r_i \in R \mid (r_i, h_j) \in E\}$ akzeptable Ärzte für $h_j \in H$,

Zuweisung von Ärzten auf Krankenhäuser

Präferenzen: beidseitig

Ziel: $(many : 1)$-Zuweisung



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

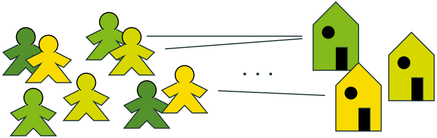
Eine HR-Instanz besteht aus

- disjunkten Agentenmengen $R = \{r_1, \ldots, r_{n_1}\}$ und $H = \{h_1, \ldots, h_{n_2}\}$,
- Kapazität c_j für jedes $h_j \in H$,
- Menge $E \subseteq R \times H$ der akzeptablen Paare, $m = \|E\|$, mit
  - $A(r_i) = \{h_j \in H \mid (r_i, h_j) \in E\}$ akzeptable Krankenhäuser für $r_i \in R$
  - $A(h_j) = \{r_i \in R \mid (r_i, h_j) \in E\}$ akzeptable Ärzte für $h_j \in H$,
- strikter Präferenzliste über $A(k)$ für jeden $a_k \in R \cup H$.

Zuweisung von Ärzten auf Krankenhäuser

Präferenzen: beidseitig

Ziel: $(\mathit{many} : 1)$-Zuweisung



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

Eine HR-Instanz besteht aus

- disjunkten Agentenmengen $R = \{r_1, \ldots, r_{n_1}\}$ und $H = \{h_1, \ldots, h_{n_2}\}$,
- Kapazität c_j für jedes $h_j \in H$,
- Menge $E \subseteq R \times H$ der akzeptablen Paare, $m = \|E\|$, mit
  - $A(r_i) = \{h_j \in H \mid (r_i, h_j) \in E\}$ akzeptable Krankenhäuser für $r_i \in R$
  - $A(h_j) = \{r_i \in R \mid (r_i, h_j) \in E\}$ akzeptable Ärzte für $h_j \in H$,
- strikter Präferenzliste über $A(k)$ für jeden $a_k \in R \cup H$.

$M \subseteq E$ ist ein *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 $\mathcal{I}$ eine HR-Instanz und M ein Matching in $\mathcal{I}$.

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

Zuweisung von Ärzten auf Krankenhäuser

Seien $\mathcal{I}$ eine HR-Instanz und M ein Matching in $\mathcal{I}$.

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

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

Zuweisung von Ärzten auf Krankenhäuser

Seien $\mathcal{I}$ eine HR-Instanz und M ein Matching in $\mathcal{I}$.

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

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

Zuweisung von Ärzten auf Krankenhäuser

Seien $\mathcal{I}$ eine HR-Instanz und M ein Matching in $\mathcal{I}$.

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

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

Ein Matching M heißt stabil, falls kein Paar M blockiert.

Zuweisung von Ärzten auf Krankenhäuser

Seien $\mathcal{I}$ eine HR-Instanz und M ein Matching in $\mathcal{I}$.

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

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

Ein Matching M heißt stabil, falls kein Paar M blockiert.

Satz

Es gibt immer ein stabiles Matching.

Zuweisung von Ärzten auf Krankenhäuser

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

Eingabe: HR-Instanz

Ausgabe: stabiles Matching M (optimal für Ärzte)

Zuweisung von Ärzten auf Krankenhäuser

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

Eingabe: HR-Instanz

Ausgabe: stabiles Matching M (optimal für Ärzte)

Nobelpreis an L. Shapley und A. Roth, 2012

Zuweisung von Ärzten auf Krankenhäuser

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

Nobelpreis an L. Shapley und A. Roth, 2012

Eingabe: HR-Instanz

Ausgabe: stabiles Matching M (optimal für Ärzte)

1. alle Ärzte frei
2. alle Krankenhäuser komplett unbesetzt

Zuweisung von Ärzten auf Krankenhäuser

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

Nobelpreis an L. Shapley und A. Roth, 2012

Eingabe: HR-Instanz

Ausgabe: stabiles Matching 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

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

Nobelpreis an L. Shapley und A. Roth, 2012

Eingabe: HR-Instanz

Ausgabe: stabiles Matching 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
5. $\quad$ Falls h vollständig besetzt:
6. $\quad\quad r' \leftarrow$ schlechtester Arzt, der bisher h zugewiesen ist
7. $\quad\quad r'$ frei

Zuweisung von Ärzten auf Krankenhäuser

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

Nobelpreis an L. Shapley und A. Roth, 2012

Eingabe: HR-Instanz

Ausgabe: stabiles Matching 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
5. $\quad$ Falls h vollständig besetzt:
6. $\quad\quad r' \leftarrow$ schlechtester Arzt, der bisher h zugewiesen ist
7. $\quad\quad r'$ frei
8. $\quad$ 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: stabiles Matching 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

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

Nobelpreis an L. Shapley und A. Roth, 2012

Eingabe: HR-Instanz

Ausgabe: stabiles Matching 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
  - Falls h vollständig besetzt:
    - $s \leftarrow$ 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: stabiles Matching 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
    Falls h vollständig besetzt:
        $s \leftarrow$ 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 das eindeutige stabiles Matching M_a aus, das aus Sicht der Ärzte optimal ist.

Zuweisung von Ärzten auf Krankenhäuser

Der oben angegebene Algorithmus gibt das eindeutige stabiles Matching M_a aus, das aus Sicht der Ärzte optimal ist.

Analog gibt es ein eindeutiges stabiles Matching M_z, das aus Sicht der Krankenhäuser optimal ist.

Zuweisung von Ärzten auf Krankenhäuser

Der oben angegebene Algorithmus gibt das eindeutige stabiles Matching M_a aus, das aus Sicht der Ärzte optimal ist.

Analog gibt es ein eindeutiges stabiles Matching M_z, das aus Sicht der Krankenhäuser optimal ist.

Dazwischen kann es noch weitere stabiles Matchings 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 das eindeutige stabiles Matching M_a aus, das aus Sicht der Ärzte optimal ist.

Analog gibt es ein eindeutiges stabiles Matching M_z, das aus Sicht der Krankenhäuser optimal ist.

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

Satz (*Rural Hospitals*)

Für eine gegebene HR*-Instanz gelten:*

Zuweisung von Ärzten auf Krankenhäuser

Der oben angegebene Algorithmus gibt das eindeutige stabiles Matching M_a aus, das aus Sicht der Ärzte optimal ist.

Analog gibt es ein eindeutiges stabiles Matching M_z, das aus Sicht der Krankenhäuser optimal ist.

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

Satz (*Rural Hospitals*)

Für eine gegebene HR*-Instanz gelten:*

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

Zuweisung von Ärzten auf Krankenhäuser

Der oben angegebene Algorithmus gibt das eindeutige stabiles Matching M_a aus, das aus Sicht der Ärzte optimal ist.

Analog gibt es ein eindeutiges stabiles Matching M_z, das aus Sicht der Krankenhäuser optimal ist.

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

Satz (*Rural Hospitals*)

Für eine gegebene HR*-Instanz gelten:*

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

Zuweisung von Ärzten auf Krankenhäuser

Der oben angegebene Algorithmus gibt das eindeutige stabiles Matching M_a aus, das aus Sicht der Ärzte optimal ist.

Analog gibt es ein eindeutiges stabiles Matching M_z, das aus Sicht der Krankenhäuser optimal ist.

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

Satz (*Rural Hospitals*)

Für eine gegebene HR*-Instanz gelten:*

- *In allen stabilen Matchings bekommen die gleichen Ärzte ein Krankenhaus zugewiesen.*
- *Jedes Krankenhaus bekommt in allen stabilen Matchings die gleiche Anzahl an Ärzten.*
- *Jedes Krankenhaus, das in einem stabilen Matching unterbesetzt ist, bekommt in jedem stabilen Matching die gleichen Ärzten zugewiesen.*

Zuweisung von Ärzten auf Krankenhäuser

Varianten:

- stabiles Heiratsproblem ($1 : 1$)

D. Manlove. **Algorithmics Of Matching Under Preferences.** Kapitel 3 und 5. Cambridge University Press, 2016.

Zuweisung von Ärzten auf Krankenhäuser

Varianten:

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

D. Manlove. **Algorithmics Of Matching Under Preferences.** Kapitel 3 und 5. Cambridge University Press, 2016.

Zuweisung von Ärzten auf Krankenhäuser

Varianten:

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

D. Manlove. **Algorithmics Of Matching Under Preferences.** Kapitel 3 und 5. Cambridge University Press, 2016.

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

D. Manlove. **Algorithmics Of Matching Under Preferences.** Kapitel 3 und 5. Cambridge University Press, 2016.

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
- ($many : many$)-Varianten: Zuweisung von Arbeitern und Firmen

D. Manlove. **Algorithmics Of Matching Under Preferences.** Kapitel 3 und 5. Cambridge University Press, 2016.

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
- (*many* : *many*)-Varianten: Zuweisung von Arbeitern und Firmen
- das Projekt-Zuweisungs-Problem
  - Kapazitäten von Projekten und Dozierenden
  - Studierende und Dozierende haben Präferenzen über Teilmengen (Studierende)
  - Angebote von Dozierenden

D. Manlove. **Algorithmics Of Matching Under Preferences.** Kapitel 3 und 5. Cambridge University Press, 2016.

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
- $(many : many)$-Varianten: Zuweisung von Arbeitern und Firmen
- das Projekt-Zuweisungs-Problem
    - Kapazitäten von Projekten und Dozierenden
    - Studierende und Dozierende haben Präferenzen über Teilmengen (Studierende)
    - Angebote von Dozierenden
- tripartites Matching
- ...

D. Manlove. **Algorithmics Of Matching Under Preferences.** Kapitel 3 und 5. Cambridge University Press, 2016.

Zuweisung von Ärzten auf Krankenhäuser

Varianten:

Thema 13

Thema 14

- stabiles Heiratsproblem (1 : 1)
- Indifferenzen in Präferenzen: schwache, starke und Super-Stabilität
- Berücksichtigung von Paaren
- Quoten: Mindestanzahlen und gemeinsame Kapazitäten
- (*many* : *many*)-Varianten: Zuweisung von Arbeitern und Firmen
- das Projekt-Zuweisungs-Problem

  Thema 15

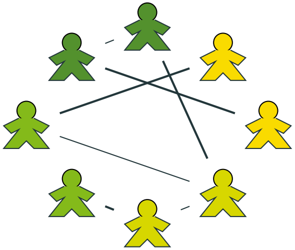
  - Kapazitäten von Projekten und Dozierenden
  - Studierende und Dozierende haben Präferenzen über Teilmengen (Studierende)
  - Angebote von Dozierenden
- tripartites Matching
- ...

D. Manlove. **Algorithmics Of Matching Under Preferences.** Kapitel 3 und 5. Cambridge University Press, 2016.

Zuordnung von Mitbewohnern

Präferenzen: nicht-bipartit, möglicherweise unvollständig, über alle möglichen Mitbewohner

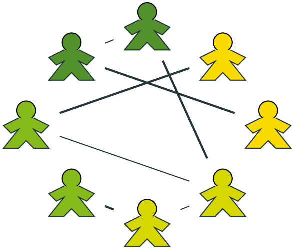
Ziel: paarweises Matching



Zuordnung von Mitbewohnern

Präferenzen: nicht-bipartit, möglicherweise unvollständig, über alle möglichen Mitbewohner

Ziel: paarweises Matching



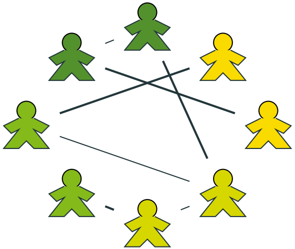
Definition (Das *Stable-Roommates*-Problem (SR))

Eine SR-Instanz besteht aus

Zuordnung von Mitbewohnern

Präferenzen: nicht-bipartit, möglicherweise unvollständig, über alle möglichen Mitbewohner

Ziel: paarweises Matching



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

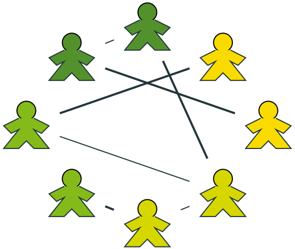
Eine SR-Instanz besteht aus

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

Zuordnung von Mitbewohnern

Präferenzen: nicht-bipartit, möglicherweise unvollständig, über alle möglichen Mitbewohner

Ziel: paarweises Matching



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

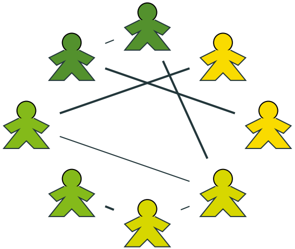
Eine SR-Instanz besteht aus

- Agenten $A = \{a_1, \ldots, a_n\}$,
- Menge $E = \{\{a_i, a_j\} \mid a_i, a_j \in A, i \neq j, \{a_i, a_j\} \text{ akzeptable Paare}\}$, $m = \|E\|$ mit $A(a_i) = \{a_j \in E \mid \{a_i, a_j\} \in E\}$ akzeptable Mitbewohner für $a_i \in A$,

Zuordnung von Mitbewohnern

Präferenzen: nicht-bipartit, möglicherweise unvollständig, über alle möglichen Mitbewohner

Ziel: paarweises Matching



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

Eine SR-Instanz besteht aus

- Agenten $A = \{a_1, \ldots, a_n\}$,
- Menge $E = \{\{a_i, a_j\} \mid a_i, a_j \in A, i \neq j, \{a_i, a_j\} \text{ akzeptable Paare}\}$, $m = \|E\|$ mit $A(a_i) = \{a_j \in E \mid \{a_i, a_j\} \in E\}$ akzeptable Mitbewohner für $a_i \in A$,
- strikter Präferenzliste über $A(a_i)$ für jeden $a_i \in A$.

Stabiles Mitbewohnerproblem

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

Stabiles Mitbewohnerproblem

Sei $\mathcal{I}$ eine $\textsc{sr}$-Instanz. $M \subseteq E$ ist eine Matching in $\mathcal{I}$, 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 $\mathcal{I}$ eine SR-Instanz. $M \subseteq E$ ist eine Matching in $\mathcal{I}$, 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 $\mathcal{I}$ eine SR-Instanz. $M \subseteq E$ ist eine Matching in $\mathcal{I}$, 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.

$\mathcal{I}$ heißt lösbar, falls es ein stabiles Matching in $\mathcal{I}$ gibt.

Stabiles Mitbewohnerproblem

Sei $\mathcal{I}$ eine SR-Instanz. $M \subseteq E$ ist eine Matching in $\mathcal{I}$, 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.

$\mathcal{I}$ heißt lösbar, falls es ein stabiles Matching in $\mathcal{I}$ gibt.

Konzepte

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

D. Manlove. **Algorithmics Of Matching Under Preferences.** Kapitel 4. Cambridge University Press, 2016.

Stabiles Mitbewohnerproblem

Sei $\mathcal{I}$ eine SR-Instanz. $M \subseteq E$ ist eine Matching in $\mathcal{I}$, 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.

$\mathcal{I}$ heißt lösbar, falls es ein stabiles Matching in $\mathcal{I}$ gibt.

Konzepte

Thema 16

Thema 17

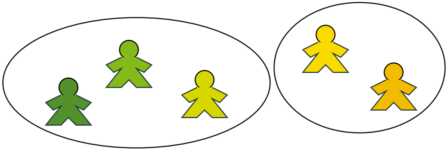
- SR-Instanzen sind nicht immer lösbar.
- Es gibt immer eine stabile Partitionierung.
- Fast-stabile Zuweisungen berücksichtigen geringe Kompromisse.

D. Manlove. **Algorithmics Of Matching Under Preferences.** Kapitel 4. Cambridge University Press, 2016.

hedonische Spiele

Präferenzen: über Koalitionen; Zufriedenheit der Spieler hängt nur von eigener Koalition ab

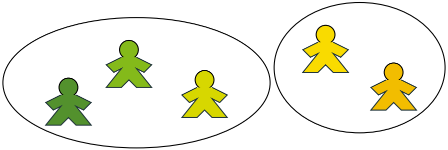
Ziel: Koalitionsbildung



hedonische Spiele

Präferenzen: über Koalitionen; Zufriedenheit der Spieler hängt nur von eigener Koalition ab

Ziel: Koalitionsbildung



Definition

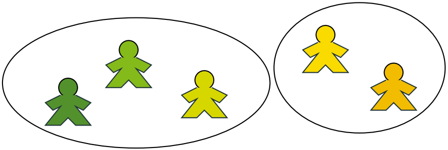
Ein hedonisches Spiel $(N, \succeq)$ besteht aus:

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

Ein Partitionierung Γ von N heißt Koalitionsstruktur. Für $i \in N$ bezeichne $\Gamma(i) \in \Gamma$ die Koalition $i \in \Gamma(i)$.

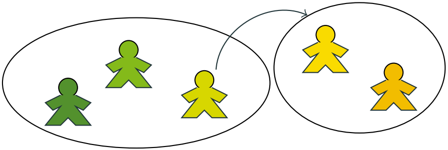
hedonische Spiele

Ziel: stabile Koalitionsstruktur finden



hedonische Spiele

Ziel: stabile Koalitionsstruktur finden



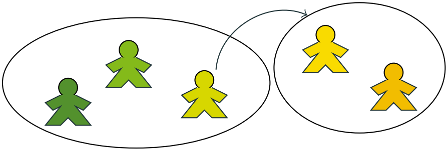
Definition

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

- Nash-stabil, wenn $\forall i \in N$, $\forall C \in \Gamma \cup \{\emptyset\}$: $\Gamma(i) \succeq C$.

hedonische Spiele

Ziel: stabile Koalitionsstruktur finden



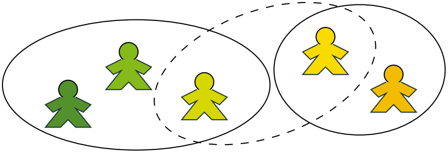
Definition

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

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

hedonische Spiele

Ziel: stabile Koalitionsstruktur finden



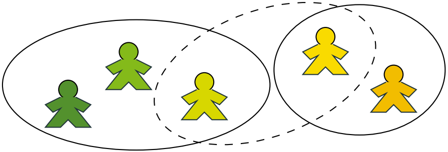
Definition

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

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

hedonische Spiele

Ziel: stabile Koalitionsstruktur finden



Definition

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

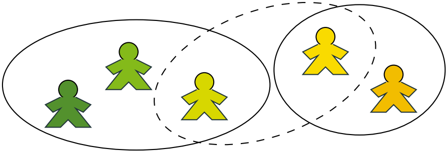
- kernstabil, wenn es keine blockierende Koalition gibt,
  d. h., $\forall C \subseteq N$, $\exists i \in C$: $\Gamma(i) \succeq C$.

Problemstellungen:

- Darstellungen: individuell rational, additiv separabel, ...

hedonische Spiele

Ziel: stabile Koalitionsstruktur finden



Definition

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

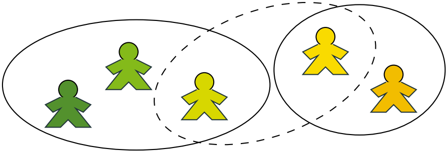
- kernstabil, wenn es keine blockierende Koalition gibt, d. h., $\forall C \subseteq N$, $\exists i \in C$: $\Gamma(i) \succeq C$.

Problemstellungen:

- Darstellungen: individuell rational, additiv separabel, ...
- *Pareto-Optimalität*: *Preference-Refinement*-Algorithmus
- Kernstabilität: *Top-Responsiveness*, *Top-Covering*-Algorithmus

H. Aziz und R. Savani. **Hedonic Games.** In Brand et al.: Handbook of Computational Social Choice, Kapitel 15. Cambridge University Press, 2016.

Ziel: stabile Koalitionsstruktur finden



Definition

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

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

Problemstellungen:

Thema 19

- Darstellungen: individuell rational, additiv separabel, ...
- *Pareto-Optimalität*: *Preference-Refinement*-Algorithmus
- Kernstabilität: *Top-Responsiveness*, *Top-Covering*-Algorithmus

Thema 18

H. Aziz und R. Savani. **Hedonic Games.** In Brand et al.: Handbook of Computational Social Choice, Kapitel 15. Cambridge University Press, 2016.

Vortragstermine I

	Thema	Datum
1	Unmöglichkeitstheoreme in Wahlverfahren	11.01.2019
2	Graphische Wahlen durch *Tournament Solutions*	11.01.2019
3	Iteratives Wählen	11.01.2019
4	*Judgment Aggregation* – Quotenregeln	11.01.2019
5	*Judgment Aggregation* – Agenda-Charakterisierung	11.01.2019
6	*Cake Cutting* – Proportionalität	18.01.2019
7	*Cake Cutting* – Neidfreiheit	18.01.2019
8	Aufteilung unteilbarer Güter – Scheidungsformel	18.01.2019
9	Aufteilung unteilbarer Güter – Approximation	18.01.2019

Vortragstermine II

	Thema	Datum
10	HA – Pareto-Optimaliät	25.01.2019
11	HA – Popularität	25.01.2019
12	HA – rang-maximale Matchings todo	25.01.2019
13	HR mit Indifferenzen – schwache Stabilität	25.01.2019
14	HR mit Indifferenzen – starke & Super-Stabilität	25.01.2019
15	Das Projekt-Zuweisungs-Problem	01.02.2019
16	SR – stabile Partitionierungen	01.02.2019
17	SR – fast-stabile Matchings	01.02.2019
18	Hedonische Spiele – Kernstabilität	01.02.2019
19	Hedonische Spiele – Pareto-Optimalität	01.02.2019

Ausblick

nächste Schritte

1. Thema auswählen bis Donnerstag
2. zugehörige Aufgabenstellung per E-Mail erhalten
3. bei Unklarheiten zeitnah nachfragen
4. beginnen!

Ausblick

nächste Schritte

1. Thema auswählen bis Donnerstag
2. zugehörige Aufgabenstellung per E-Mail erhalten
3. bei Unklarheiten zeitnah nachfragen
4. beginnen!

nächste Stunde

- Präsentationskurs am (A) 19. Oktober oder (B) 26. Oktober
- u. a.: schriftliches Präsentieren
- Unterlagen zum Proseminarthema und ggf. Notebook mitbringen