Julius-Maximilians-
UNIVERSITÄT
WÜRZBURG

Approximationsalgorithmen für Netzwerkdesign und Standortplanung

Joachim Spoerhase

Lehrstuhl für Informatik I, Universität Würzburg

Juli, 2017

Approximationsalgorithmen

Viele kombinatorische Optimierungsprobleme (z.B. Tourenplanung, Standortplanung, Scheduling) sind NP-schwer.

Approximationsalgorithmen

Viele kombinatorische Optimierungsprobleme (z.B. Tourenplanung, Standortplanung, Scheduling) sind NP-schwer.

Es gibt somit (außer P=NP) keinen effizienten Algorithmen, der eine optimale Lösung berechnet.

Approximationsalgorithmen

Viele kombinatorische Optimierungsprobleme (z.B. Tourenplanung, Standortplanung, Scheduling) sind NP-schwer.

Es gibt somit (außer P=NP) keinen effizienten Algorithmen, der eine optimale Lösung berechnet.

Ein *Approximationsalgorithmus* ist ein effizienter Algorithmus, der (beweisbar) immer eine Lösung berechnet, die um einen beschränkten Faktor schlechter ist als die optimale Lösung.

Approximationsalgorithmen

Viele kombinatorische Optimierungsprobleme (z.B. Tourenplanung, Standortplanung, Scheduling) sind NP-schwer.

Es gibt somit (außer P=NP) keinen effizienten Algorithmen, der eine optimale Lösung berechnet.

Ein *Approximationsalgorithmus* ist ein effizienter Algorithmus, der (beweisbar) immer eine Lösung berechnet, die um einen beschränkten Faktor schlechter ist als die optimale Lösung.

In dieser Arbeit Probleme aus den Bereichen **Netzwerkdesign** und **Standortplanung**.

Approximationsalgorithmen

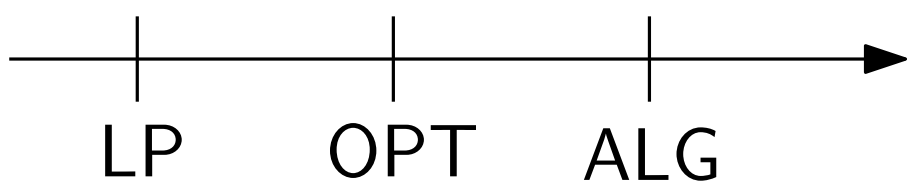
Viele kombinatorische Optimierungsprobleme (z.B. Tourenplanung, Standortplanung, Scheduling) sind NP-schwer.

Es gibt somit (außer P=NP) keinen effizienten Algorithmen, der eine optimale Lösung berechnet.

Ein *Approximationsalgorithmus* ist ein effizienter Algorithmus, der (beweisbar) immer eine Lösung berechnet, die um einen beschränkten Faktor schlechter ist als die optimale Lösung.

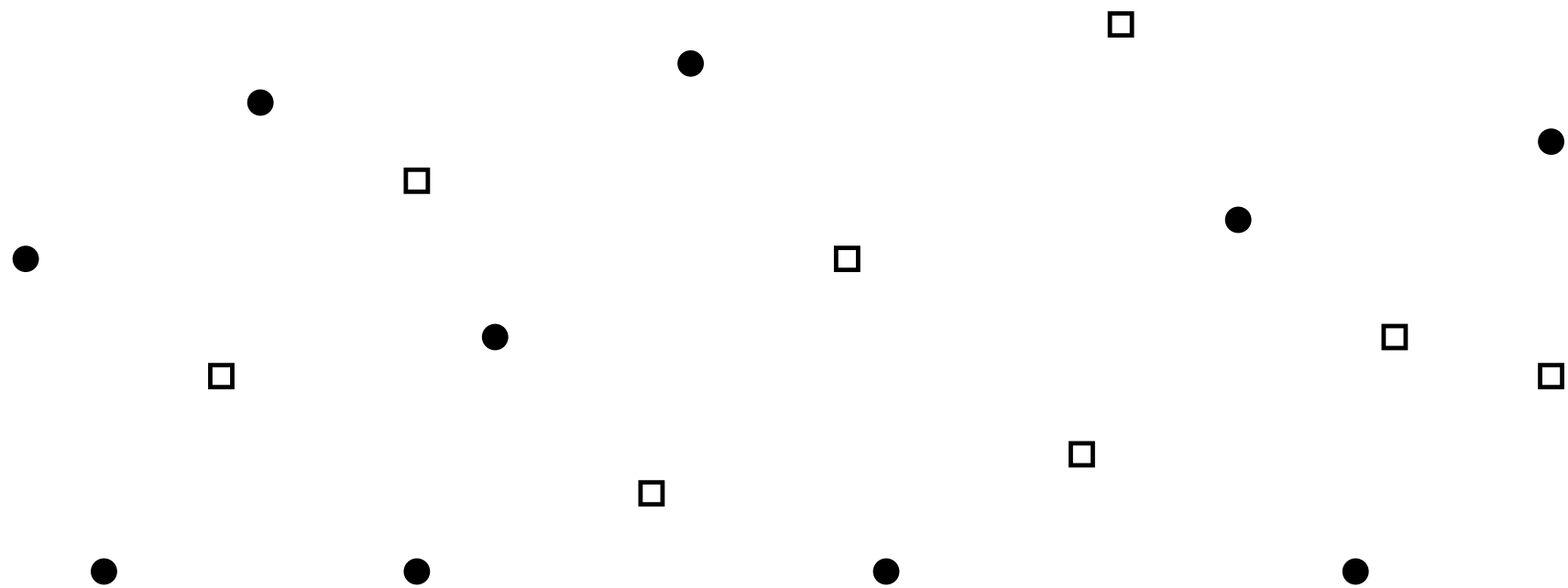
In dieser Arbeit Probleme aus den Bereichen **Netzwerkdesign** und **Standortplanung**.

Relaxierungen mittels linearer Programmierung sind wichtiges Werkzeug.



k-Median

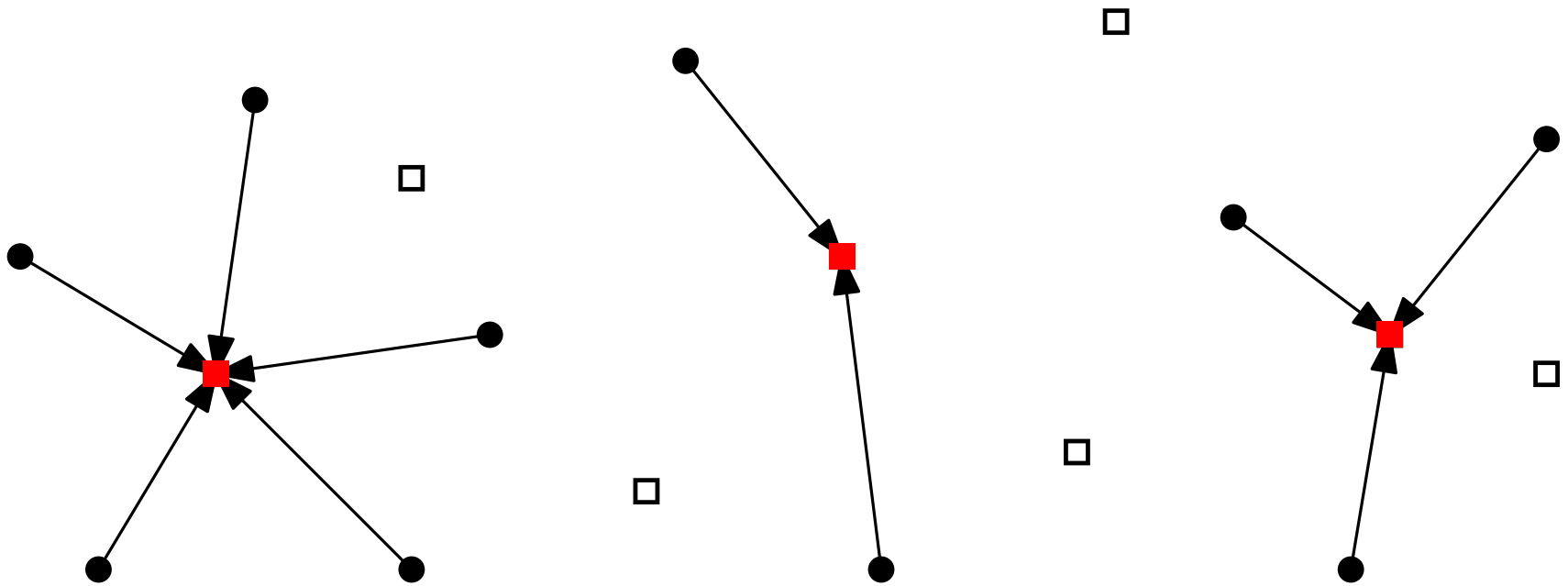
Eingabe Endlicher metrischer Raum, der in Menge F von Facilities und eine Menge C von Clients partitioniert ist, sowie eine natürliche Zahl k



k-Median

Eingabe Endlicher metrischer Raum, der in Menge F von Facilities und eine Menge C von Clients partitioniert ist, sowie eine natürliche Zahl k

Ziel Öffne k Facilities, so dass die Abstandssumme der Clients zur nächsten geöffneten Facility minimal ist



k-Median

Anwendungen: Standortplanung/Logistik, Clustering

k-Median

Anwendungen: Standortplanung/Logistik, Clustering

zentrales NP-schweres Problem im Bereich Approximationsalgorithmen mit langer Serie von Verbesserungen...

k-Median

Anwendungen: Standortplanung/Logistik, Clustering

zentrales NP-schweres Problem im Bereich Approximationsalgorithmen mit langer Serie von Verbesserungen...

Faktor	Autoren	Technik	Konferenz
$6\frac{1}{2}$	Charikar et al.	LP rounding	[STOC'99]
6	Jain, Vazirani	primal dual	[FOCS'99]
4	Jain et al.	primal dual	[STOC'02]
$3+\epsilon$	Arya et al.	local search	[STOC'01]
2.732	Li, Svensson	primal dual	[STOC'13]
2.675	Byrka et. al.	primal dual	[SODA'15]

k-Median

Anwendungen: Standortplanung/Logistik, Clustering

zentrales NP-schweres Problem im Bereich Approximationsalgorithmen mit langer Serie von Verbesserungen...

Faktor	Autoren	Technik	Konferenz
$6\frac{1}{2}$	Charikar et al.	LP rounding	[STOC'99]
6	Jain, Vazirani	primal dual	[FOCS'99]
4	Jain et al.	primal dual	[STOC'02]
$3+\epsilon$	Arya et al.	local search	[STOC'01]
2.732	Li, Svensson	primal dual	[STOC'13]
2.675	Byrka et. al.	primal dual	[SODA'15]

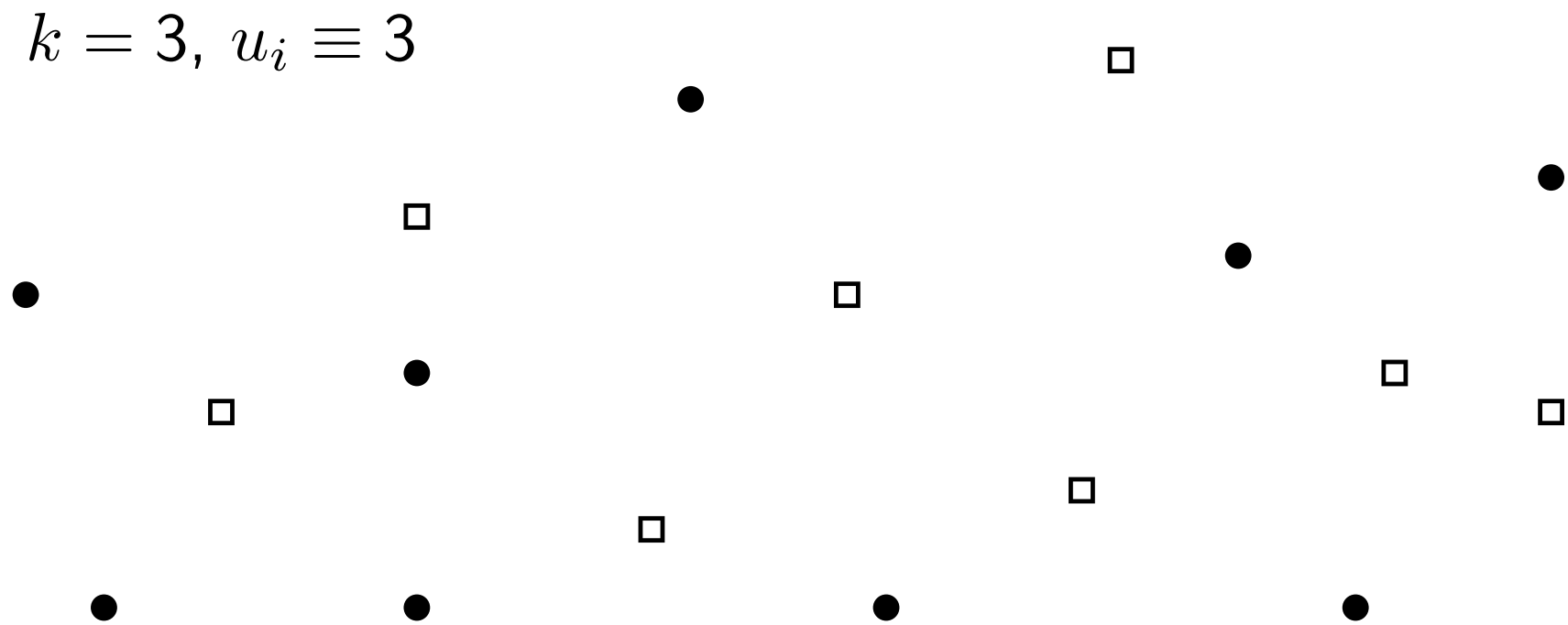
stärkste untere Schranke von 1.735 von Jain et al. [STOC'02]

Kapazitiertes k-Median

. . . nun hat jede Facility i eine individuelle Kapazität u_i

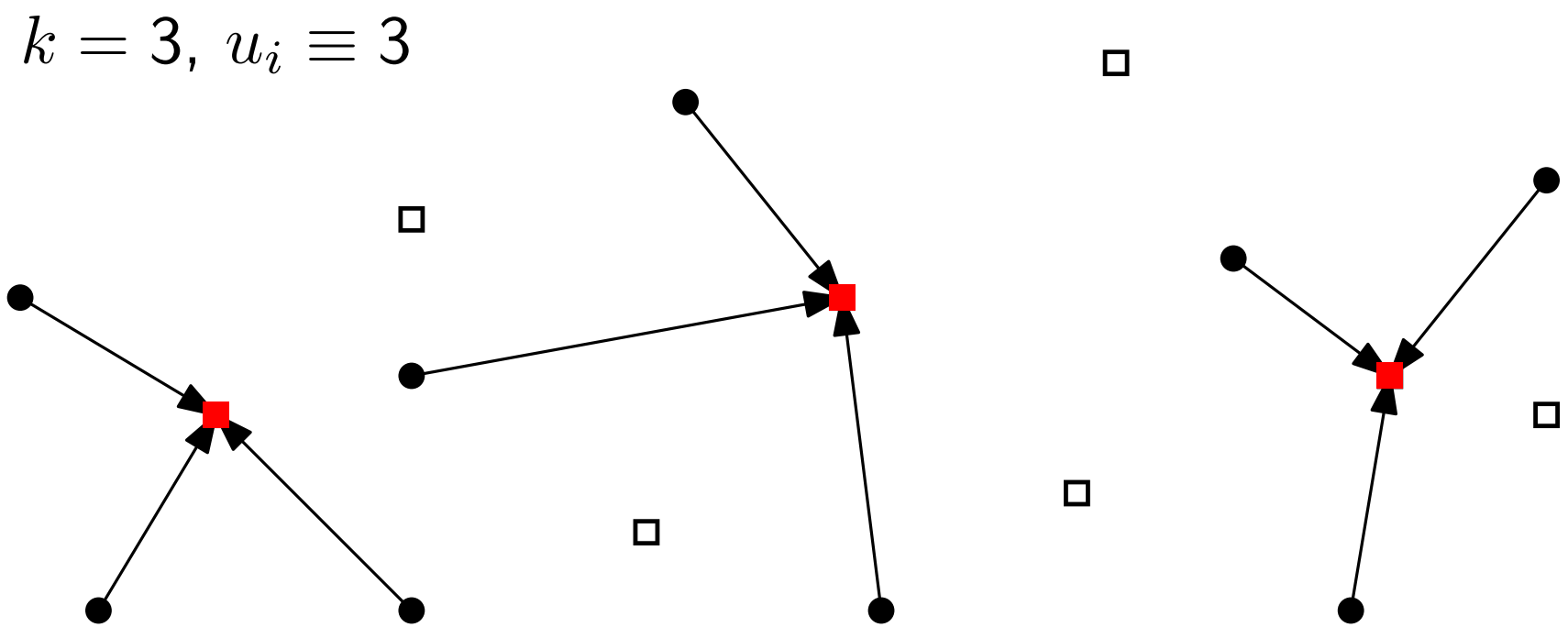
Kapazitiertes k-Median

. . . nun hat jede Facility i eine individuelle Kapazität u_i



Kapazitiertes k-Median

. . . nun hat jede Facility i eine individuelle Kapazität u_i

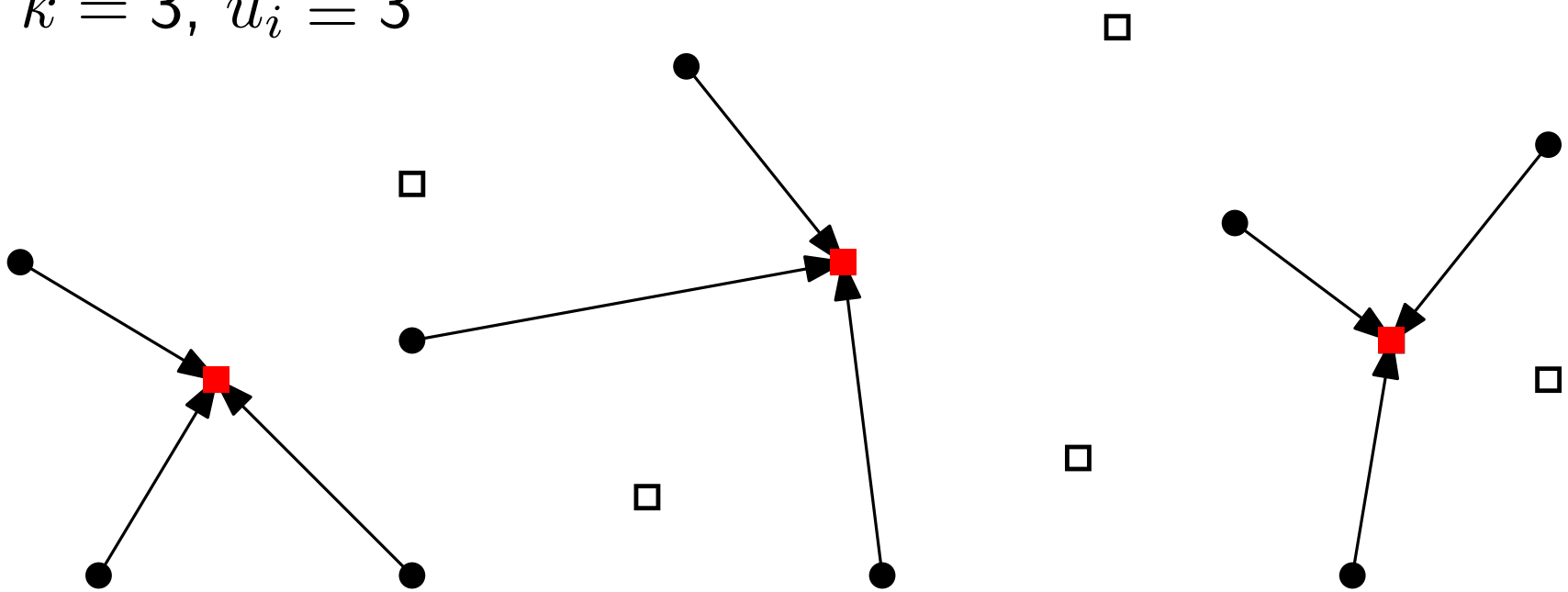


Kapazitiertes k-Median

. . . nun hat jede Facility i eine individuelle Kapazität u_i

Zuweisungsproblem bereits nicht-trivial (b-matching)

$k = 3$, $u_i \equiv 3$



Kapazitiertes k-Median

Zentrale offene Frage: Gibt es einen $O(1)$-Approximationsalgorithmus?

Kapazitiertes k-Median

Zentrale offene Frage: Gibt es einen $O(1)$-Approximationsalgorithmus?

Alle bekannten Algorithmen verletzen die Kapazitäten.

Kapazitiertes k-Median

Zentrale offene Frage: Gibt es einen $O(1)$-Approximationsalgorithmus?

Alle bekannten Algorithmen verletzen die Kapazitäten.

Faktor $(4, 16)$ für uniforme (weiche) Kapazitäten durch Charikar et al. [STOC'99].

Kapazitiertes k-Median

Zentrale offene Frage: Gibt es einen $O(1)$-Approximationsalgorithmus?

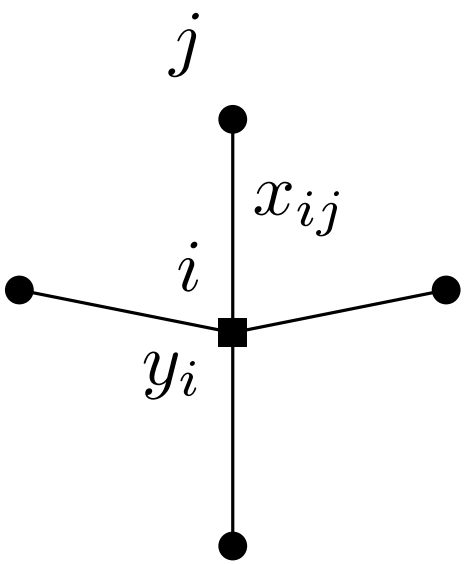
Alle bekannten Algorithmen verletzen die Kapazitäten.

Faktor $(4, 16)$ für uniforme (weiche) Kapazitäten durch Charikar et al. [STOC'99].

Faktor $(40, 50)$ für nicht-uniforme (weiche) Kapazitäten durch Chuzhoy und Rabani [SODA'04].

Standard LP-Relaxierung

$$\text{min} \sum_{i \in F, j \in C} c_{ij} x_{ij}, \quad \text{s.t.}$$

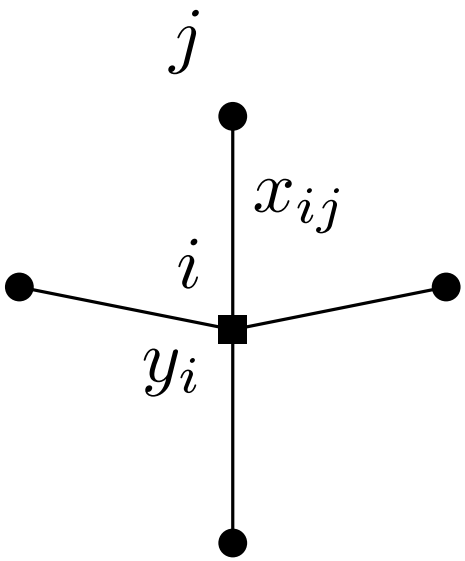


$$x_{ij}, y_i \geq 0 \qquad \forall i \in F, j \in C$$

Standard LP-Relaxierung

$$\text{min} \sum_{i \in F, j \in C} c_{ij} x_{ij}, \quad \text{s.t.}$$

$$\sum_{i \in F} x_{ij} \geq 1 \qquad \forall j \in C$$



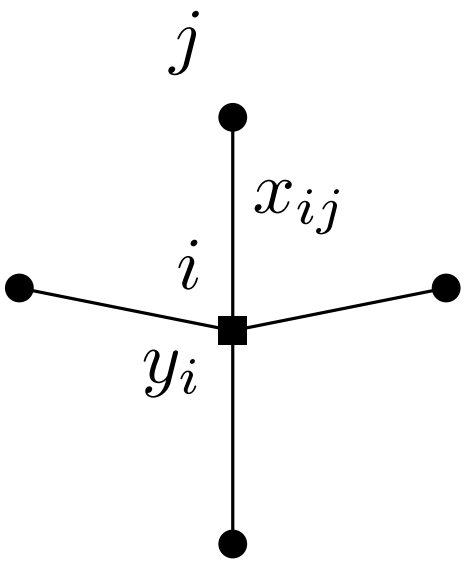
$$x_{ij}, y_i \geq 0 \qquad \forall i \in F, j \in C$$

Standard LP-Relaxierung

$$\text{min} \sum_{i \in F, j \in C} c_{ij} x_{ij}, \quad \text{s.t.}$$

$$\sum_{i \in F} x_{ij} \geq 1 \qquad \forall j \in C$$

$$x_{ij} \leq y_i \qquad \forall i \in F, j \in C$$



$$x_{ij}, y_i \geq 0 \qquad \forall i \in F, j \in C$$

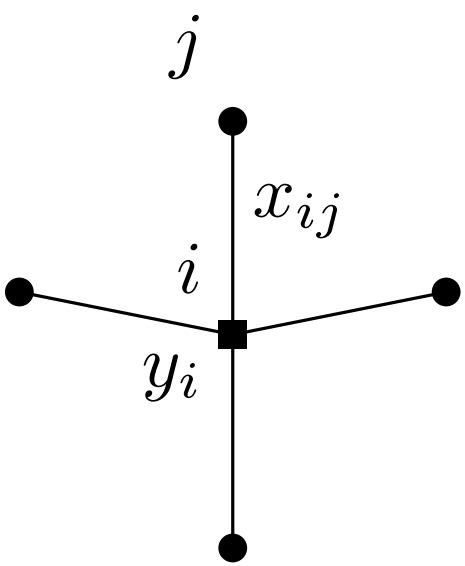
Standard LP-Relaxierung

$$\text{min} \sum_{i \in F, j \in C} c_{ij} x_{ij}, \quad \text{s.t.}$$

$$\sum_{i \in F} x_{ij} \geq 1 \qquad \forall j \in C$$

$$x_{ij} \leq y_i \qquad \forall i \in F, j \in C$$

$$\sum_{j \in C} x_{ij} \leq u_i y_i \qquad \forall i \in F$$



$$x_{ij}, y_i \geq 0 \qquad \forall i \in F, j \in C$$

Standard LP-Relaxierung

$$\text{min} \sum_{i \in F, j \in C} c_{ij} x_{ij}, \quad \text{s.t.}$$

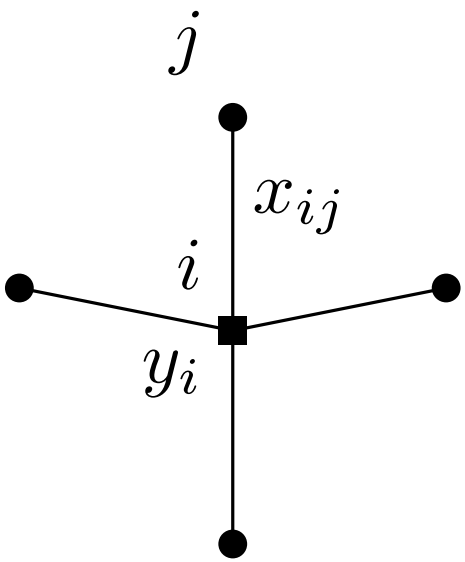
$$\sum_{i \in F} x_{ij} \geq 1 \qquad \forall j \in C$$

$$x_{ij} \leq y_i \qquad \forall i \in F, j \in C$$

$$\sum_{j \in C} x_{ij} \leq u_i y_i \qquad \forall i \in F$$

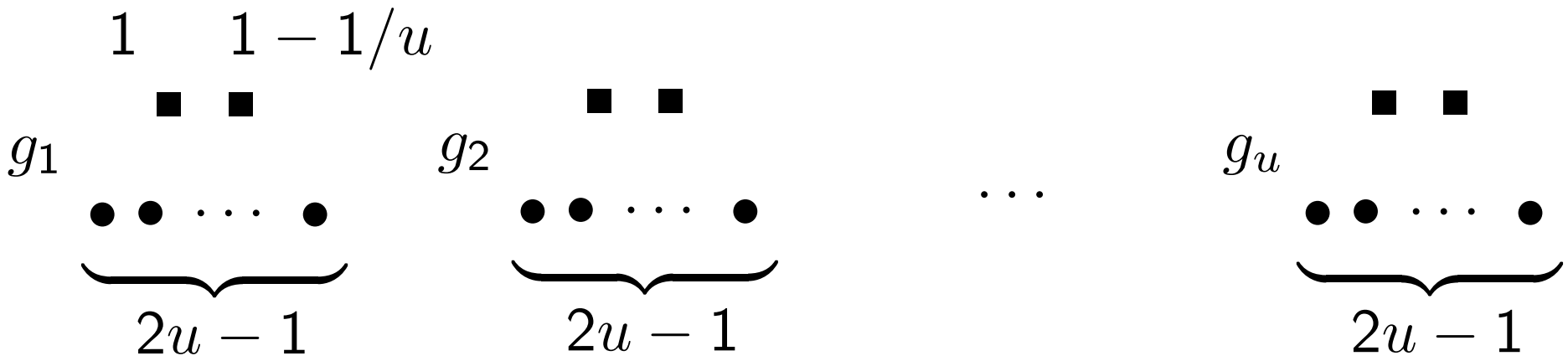
$$\sum_{i \in F} y_i \leq k$$

$$x_{ij}, y_i \geq 0 \qquad \forall i \in F, j \in C$$



Integrality-Gap

$k = 2u - 1$



Integrality-Gap unbeschränkt, außer wir verletzen die Kapazitäten um den Faktor 2 !

$$\min \sum_{i \in F, j \in C} c_{ij} x_{ij}, \quad \text{s.t.}$$

$$\sum_{i \in F} x_{ij} \geq 1 \qquad \forall j \in C$$

$$x_{ij} \leq y_i \qquad \forall i \in F, j \in C$$

$$\sum_{j \in C} x_{ij} \leq u_i y_i \qquad \forall i \in F$$

$$\sum_{i \in F} y_i \leq k$$

$$x_{ij}, y_i \geq 0 \qquad \forall i \in F, j \in C$$

Unser Resultat

Satz Es gibt einen $(2+\epsilon, O(1/\epsilon^2))$-Approximationsalgorithmus für k-Median with uniformen and harten Kapazitäten und einen $(3+\epsilon, O(1/\epsilon^2))$-Approximationsalgorithmus für harte nicht-uniforme Kapazitäten.

gemeinsame Arbeit mit Jarek Byrka, Krzystzof Fleszar, and Bartek Rybicky [SODA'15]

Unser Resultat

Satz Es gibt einen $(2+\epsilon, O(1/\epsilon^2))$-Approximationsalgorithmus für k-Median with uniformen and harten Kapazitäten und einen $(3+\epsilon, O(1/\epsilon^2))$-Approximationsalgorithmus für harte nicht-uniforme Kapazitäten.

gemeinsame Arbeit mit Jarek Byrka, Krzystzof Fleszar, and Bartek Rybicky [SODA'15]

wichtige Techniken: (Meta-)clustering

Unser Resultat

Satz Es gibt einen $(2+\epsilon, O(1/\epsilon^2))$-Approximationsalgorithmus für k-Median with uniformen and harten Kapazitäten und einen $(3+\epsilon, O(1/\epsilon^2))$-Approximationsalgorithmus für harte nicht-uniforme Kapazitäten.

gemeinsame Arbeit mit Jarek Byrka, Krzystzof Fleszar, and Bartek Rybicky [SODA'15]

wichtige Techniken: (Meta-)clustering

abhängiges, randomisiertes Runden

Unser Resultat

Satz Es gibt einen $(2+\epsilon, O(1/\epsilon^2))$-Approximationsalgorithmus für k-Median with uniformen and harten Kapazitäten und einen $(3+\epsilon, O(1/\epsilon^2))$-Approximationsalgorithmus für harte nicht-uniforme Kapazitäten.

gemeinsame Arbeit mit Jarek Byrka, Krzystzof Fleszar, and Bartek Rybicky [SODA'15]

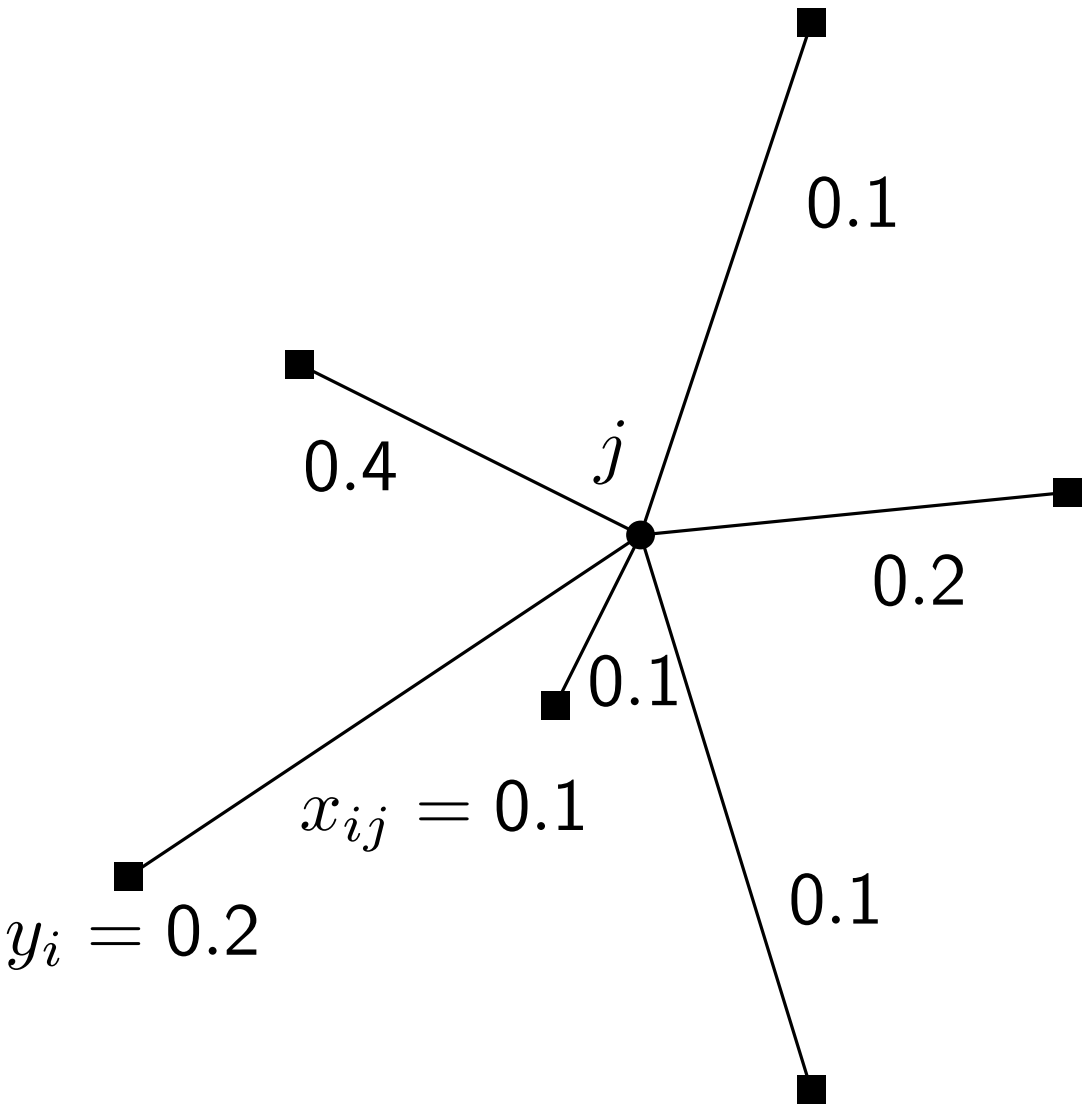
wichtige Techniken: (Meta-)clustering

abhängiges, randomisiertes Runden

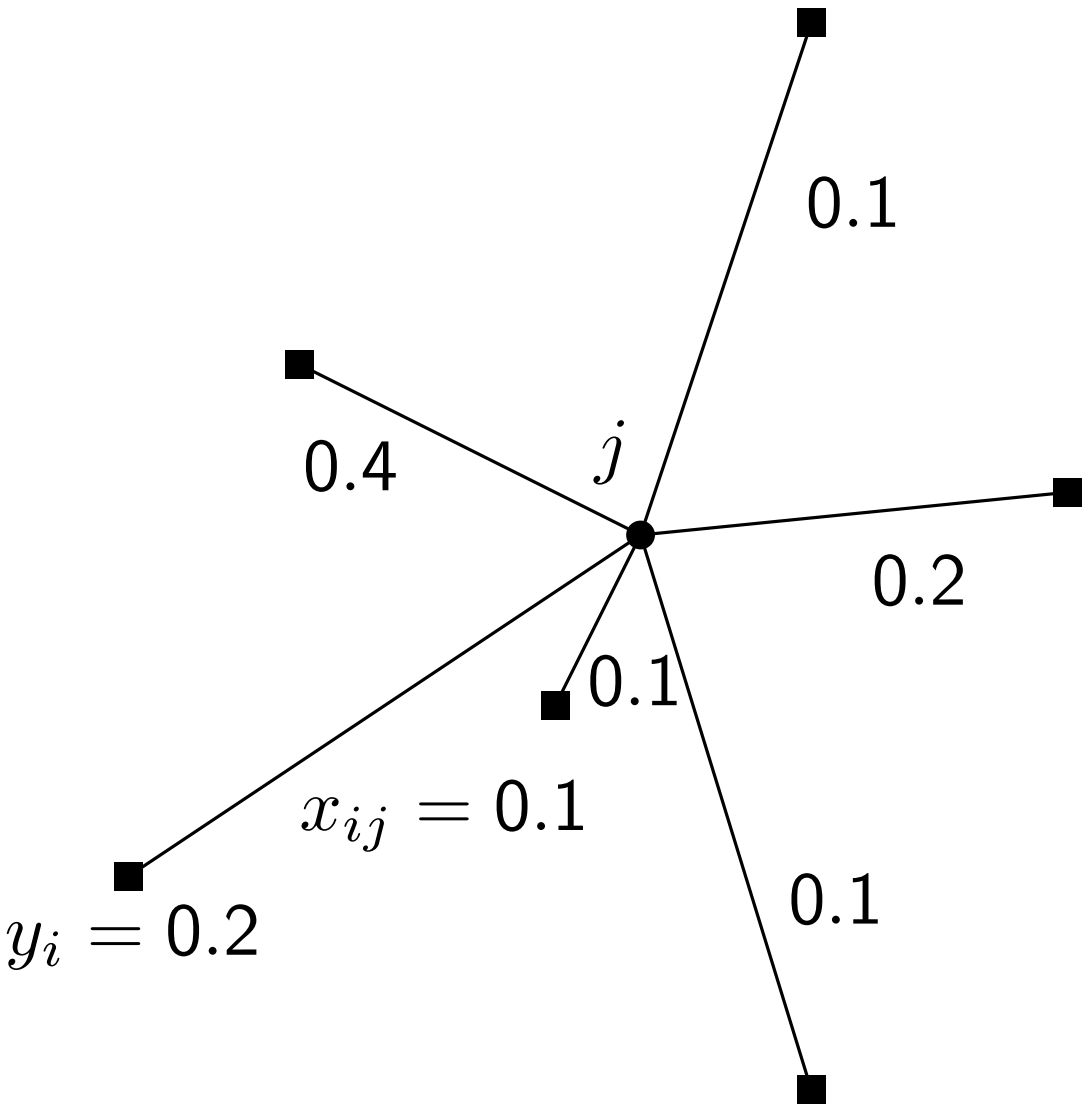
Ausnutzen der Struktur von Extrempunktlösungen

Durchschnittsdistanzen

$$d_{\mathsf{av}}(j) := \sum_{i \in F} c_{ij} x_{ij}$$



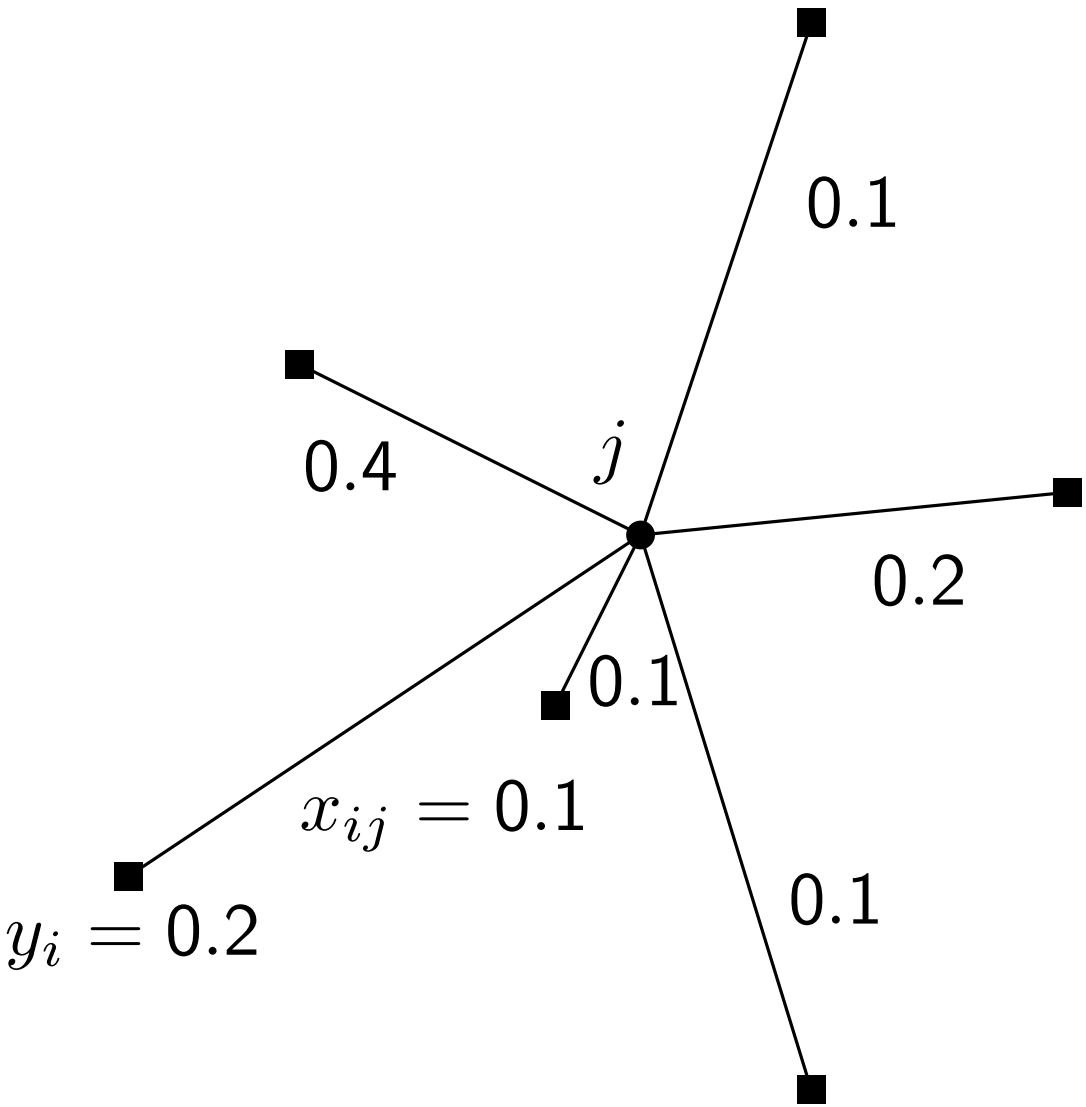
Durchschnittsdistanzen



$d_{\mathsf{av}}(j) := \sum_{i \in F} c_{ij} x_{ij}$

$\mathsf{OPT}^* = \sum_{j \in C} d_{\mathsf{av}}(j)$

Durchschnittsdistanzen



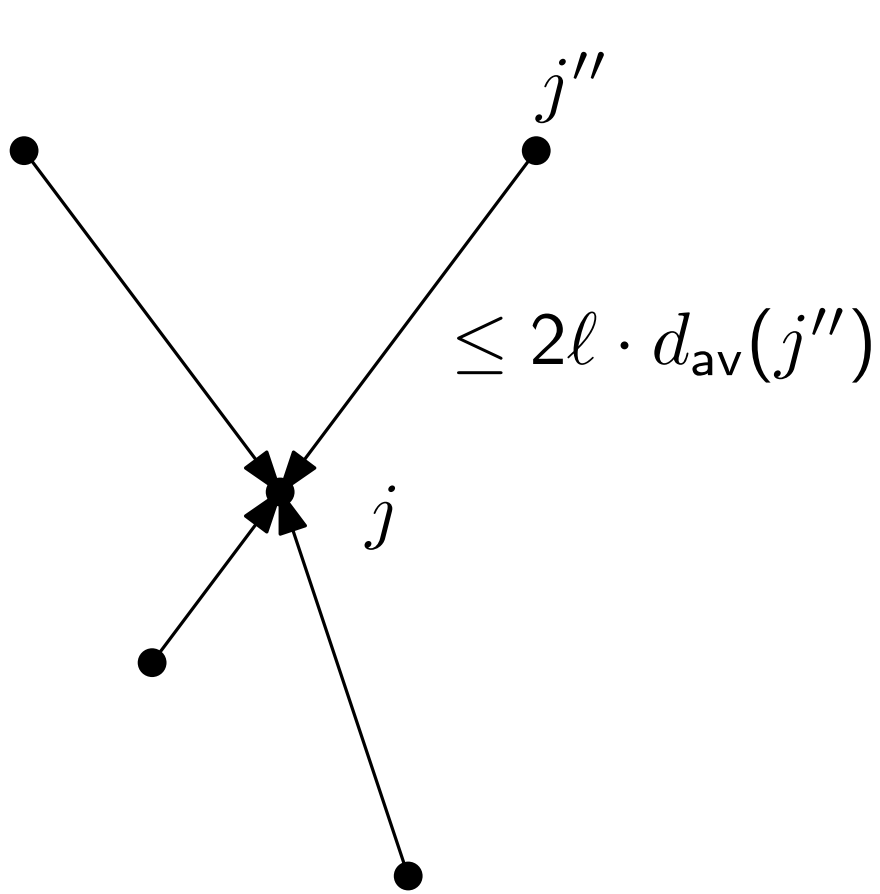
$d_{\mathsf{av}}(j) := \sum_{i \in F} c_{ij} x_{ij}$

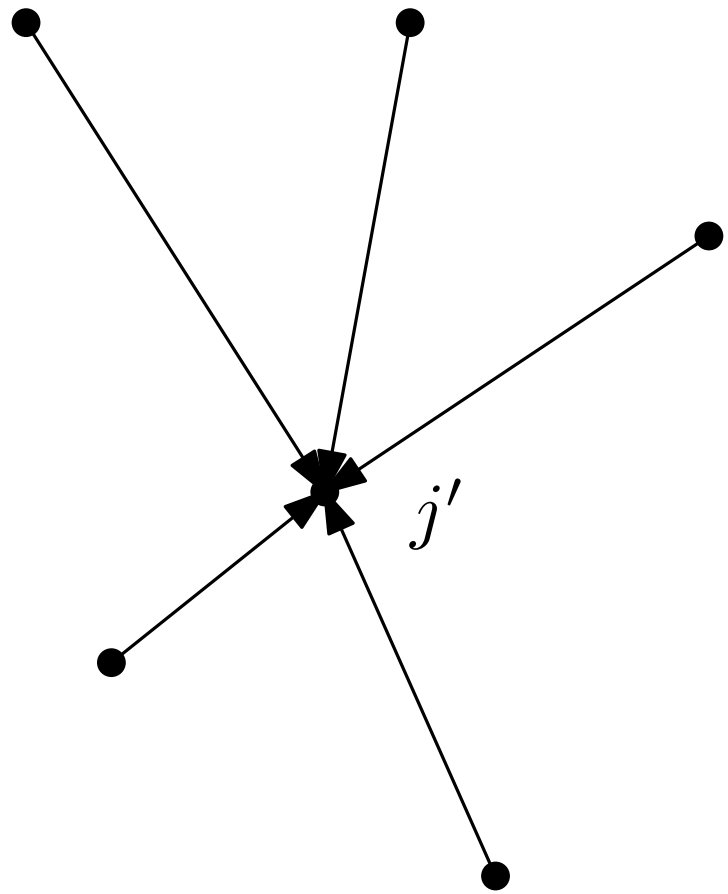
$\mathsf{OPT}^* = \sum_{j \in C} d_{\mathsf{av}}(j)$

idealerweise sind die Verbindungskosten von $j \in C$ im Algorithmus $O(d_{\mathsf{av}}(j))$

Aggregation von Clients

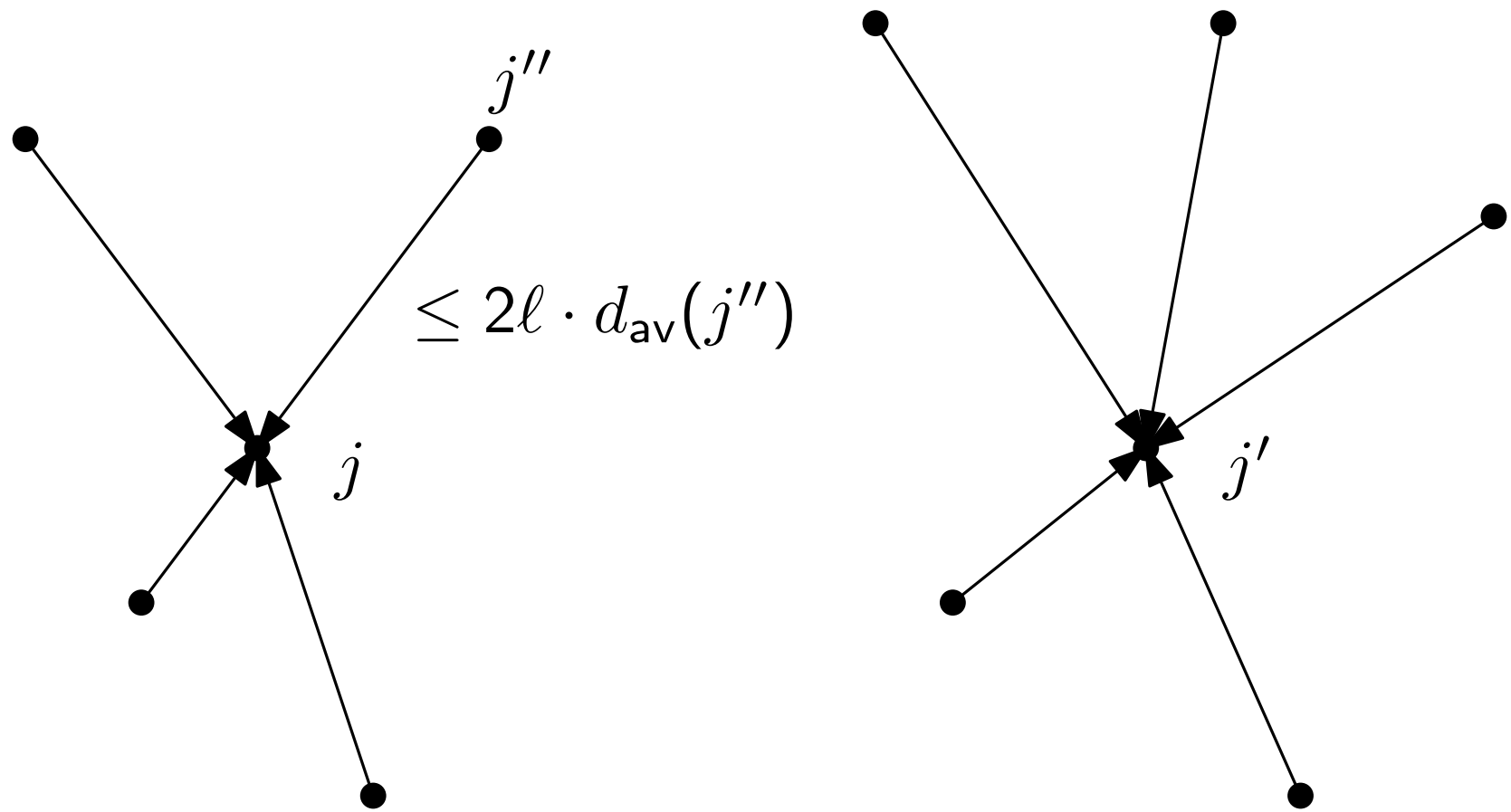
Parameter $\ell \geq 2$





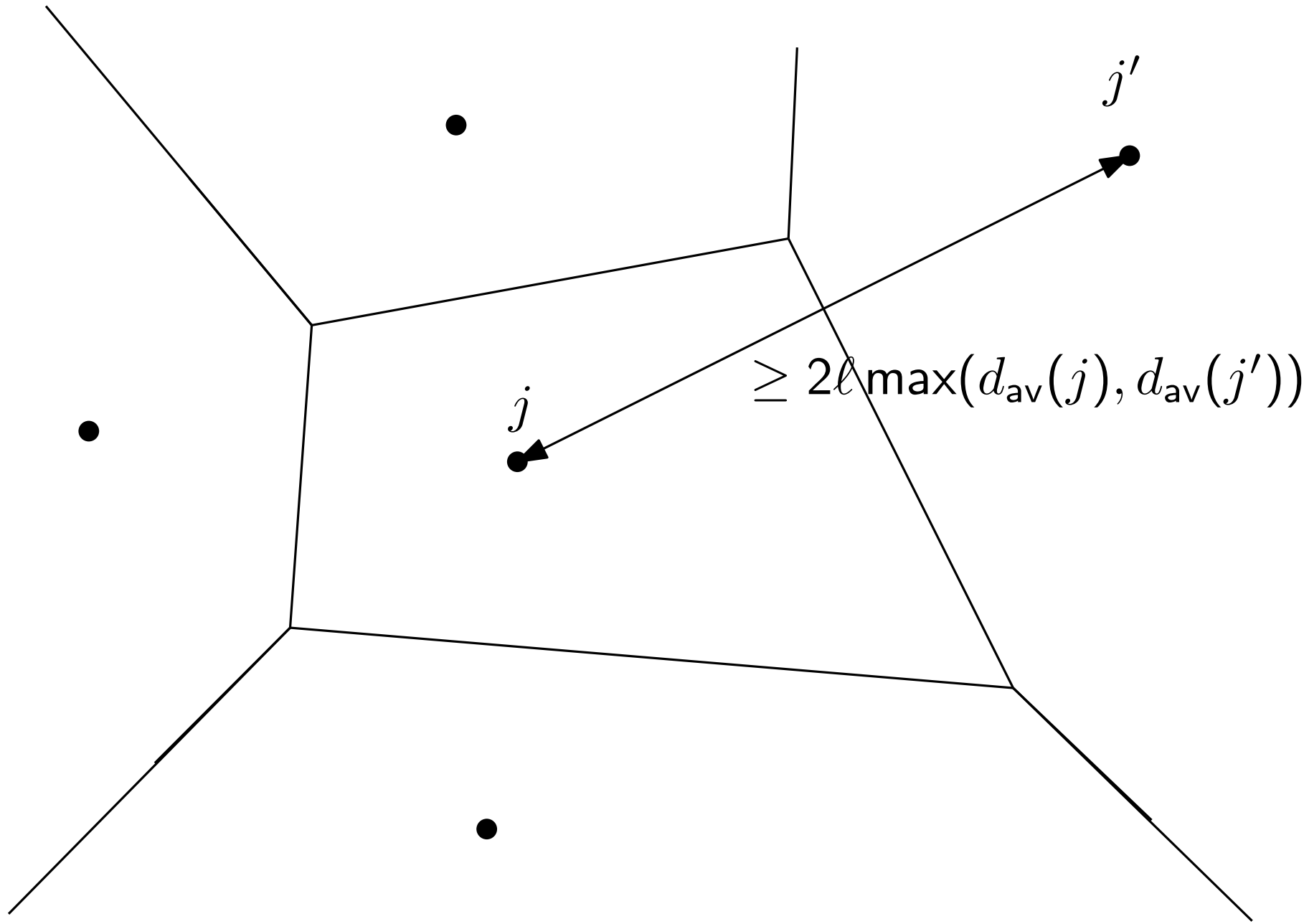
Aggregation von Clients

Parameter $\ell \geq 2$

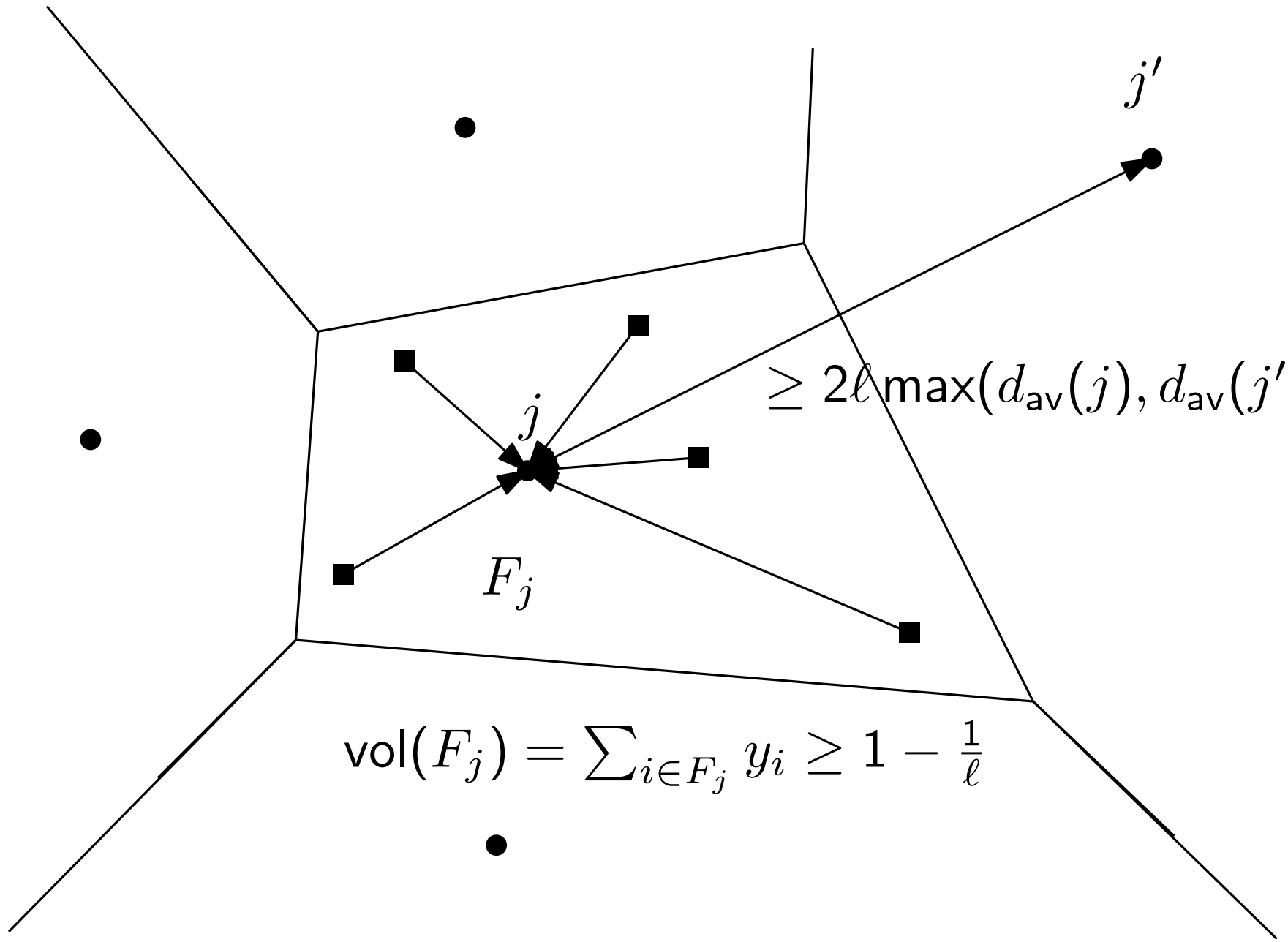


durchlaufe Clients nach aufsteigendem Durschnittsabstand und aggregiere

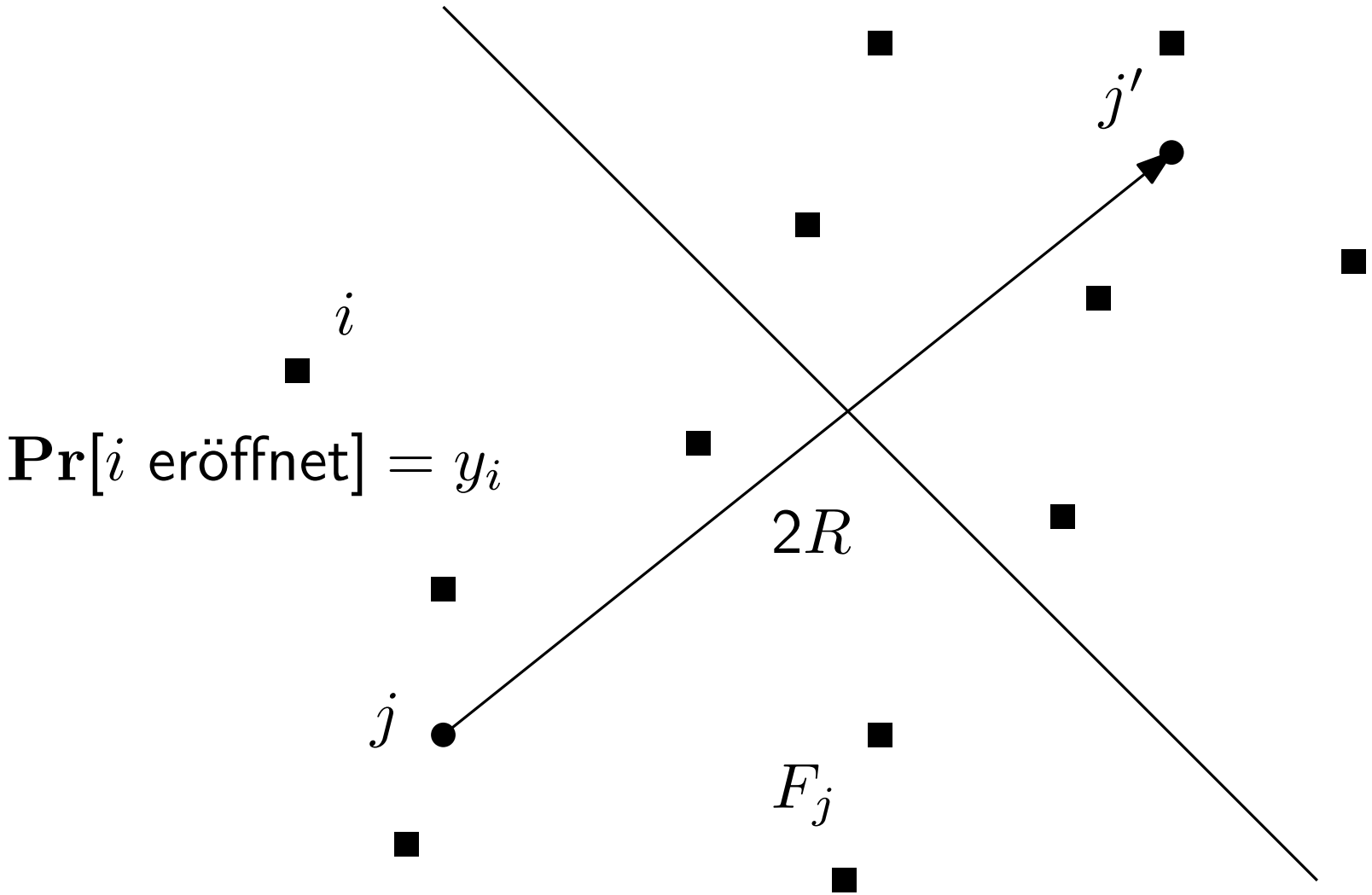
Clustering der Facilities



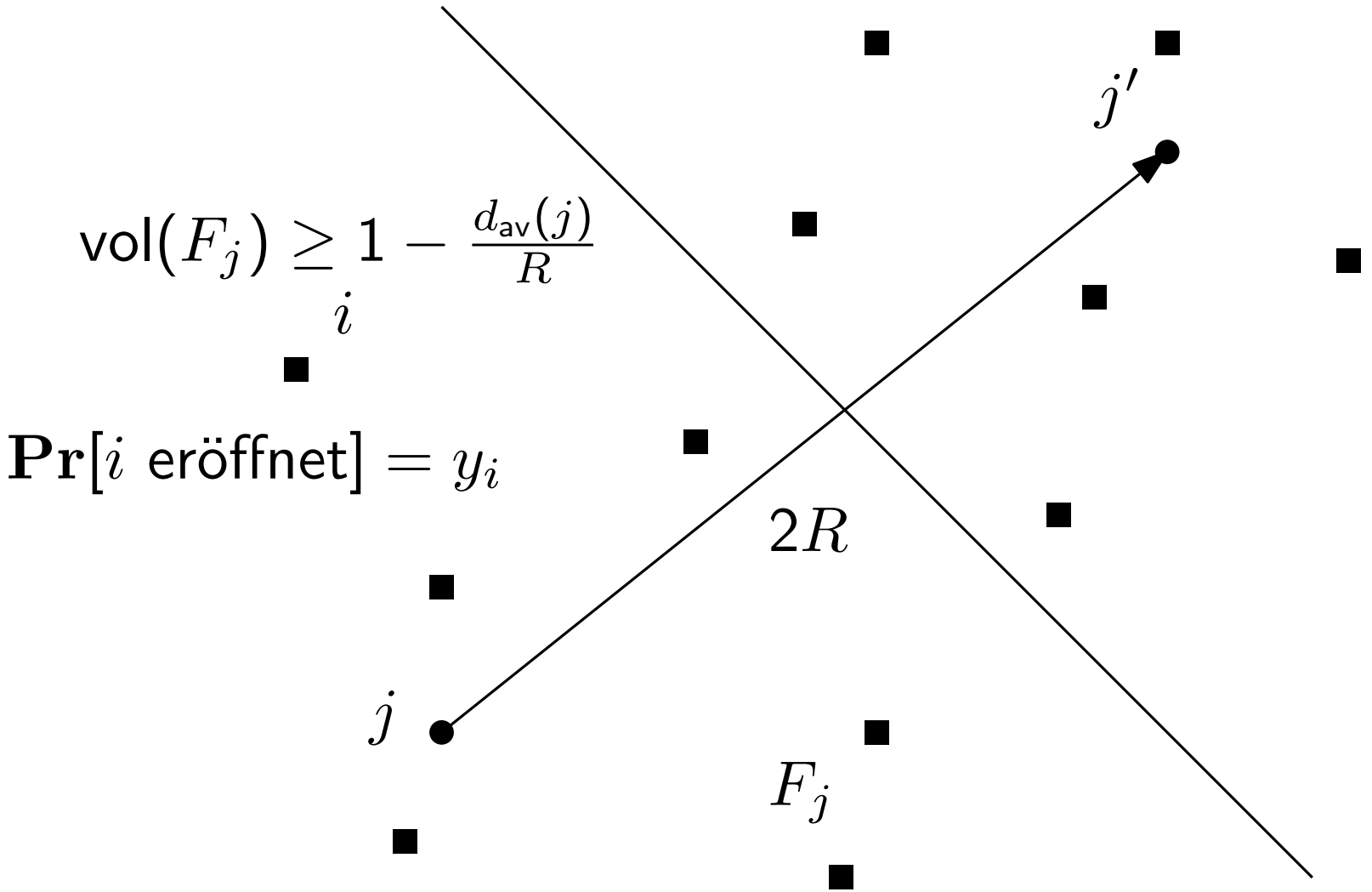
Clustering der Facilities



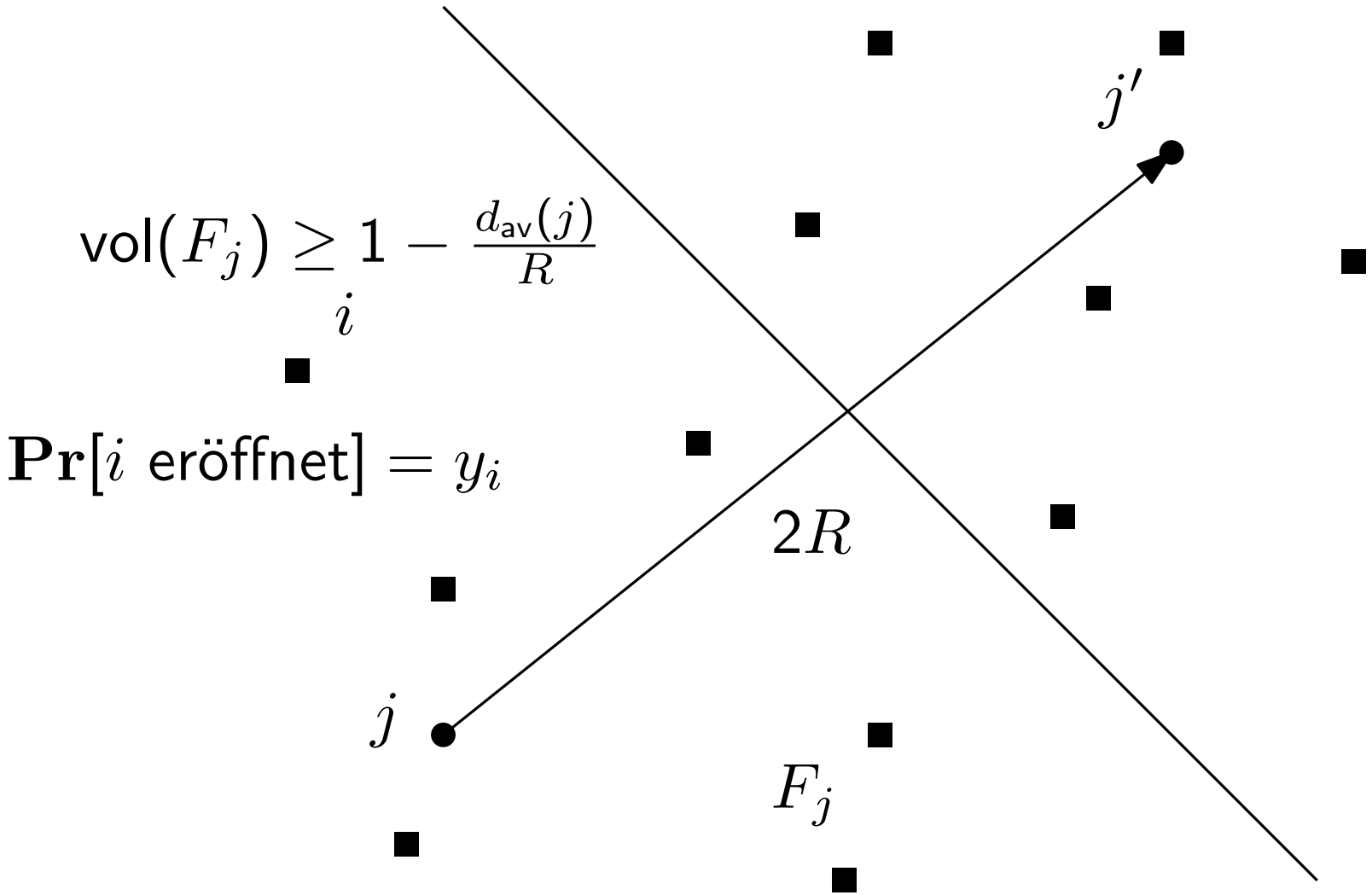
Randomisiertes Runden – Verbindungskosten



Randomisiertes Runden – Verbindungskosten

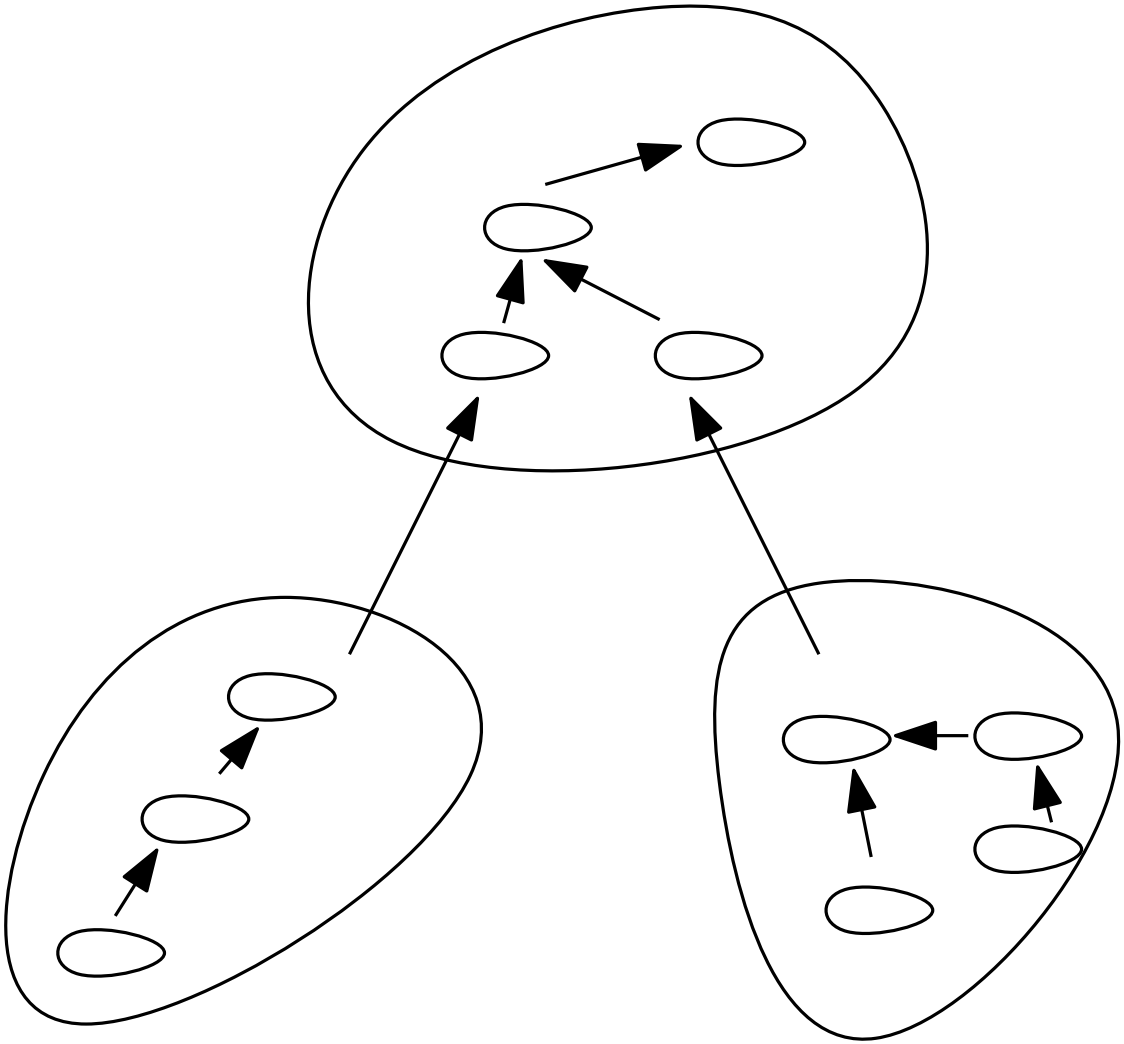


Randomisiertes Runden – Verbindungskosten



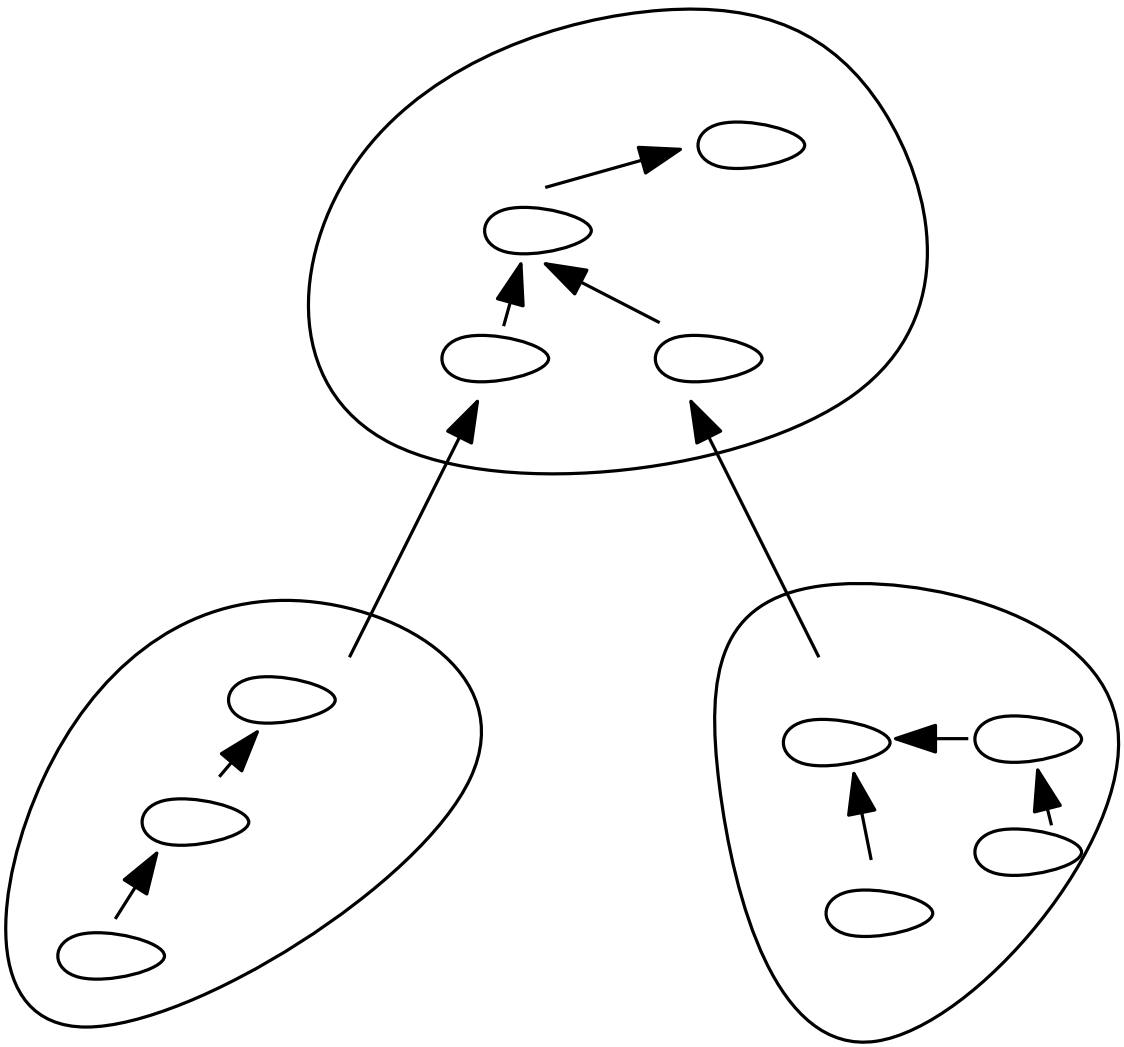
$$\mathbb{E}[\mathsf{connect}(j)] \leq d_{\mathsf{av}}(j) + \frac{d_{\mathsf{av}}(j)}{R} \cdot O(R) = O(d_{\mathsf{av}}(j))$$

Hierarchische Metacluster – Kapazitäten



Metaclustergröße$=\Theta(\ell)$

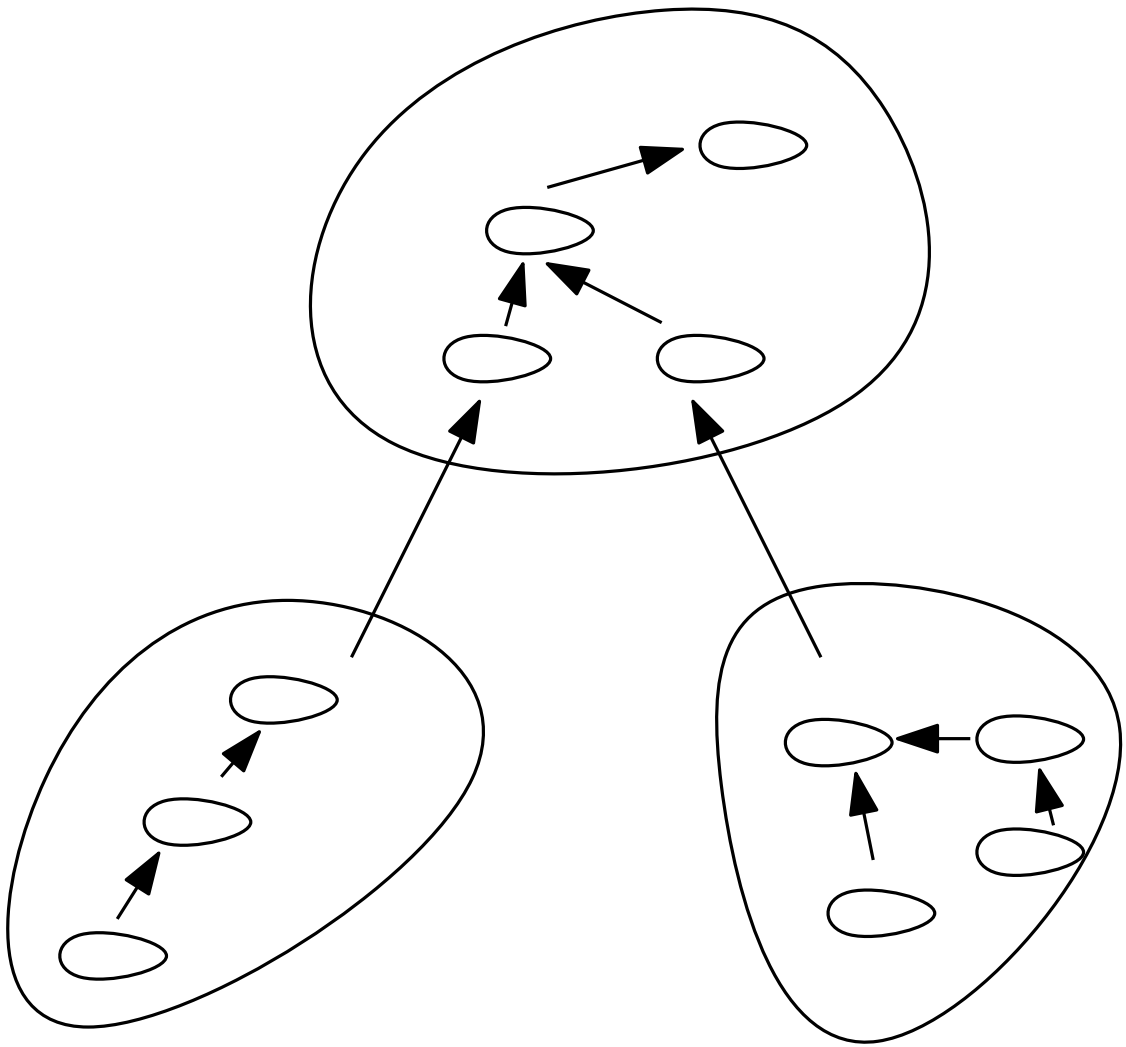
Hierarchische Metacluster – Kapazitäten



„abhängiges“ randomisiertes Runden garantiert, dass Metaclustervolumen höchstens auf- oder abgerundet wird

Metaclustergröße$=\Theta(\ell)$

Hierarchische Metacluster – Kapazitäten

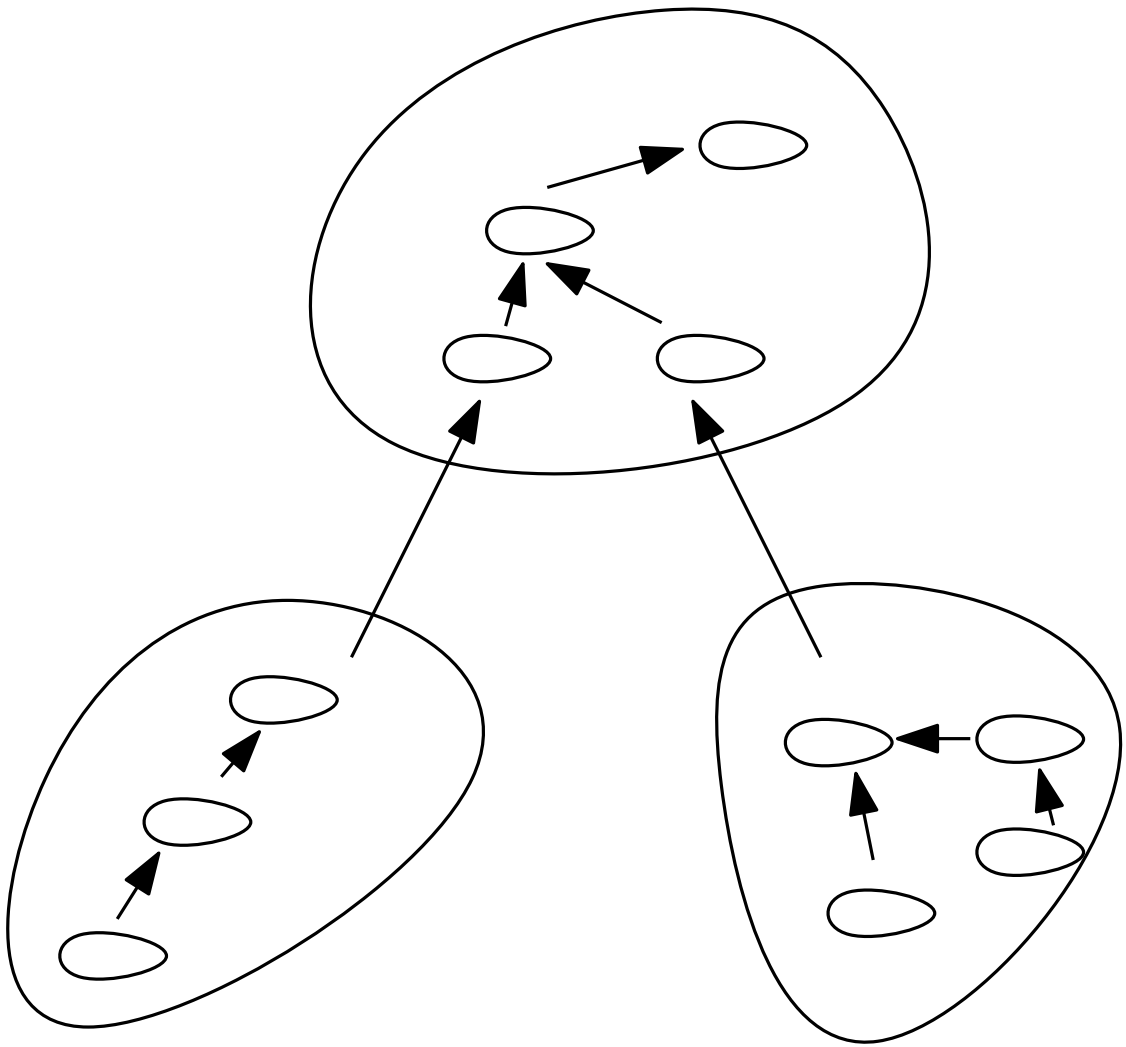


„abhängiges“ randomisiertes Runden garantiert, dass Metaclustervolumen höchstens auf- oder abgerundet wird

Kapazitätsverletzung $\approx \frac{(2-\delta)u}{\lfloor 2-\delta \rfloor u} + \frac{1}{\ell}$

Metaclustergröße$=\Theta(\ell)$

Hierarchische Metacluster – Kapazitäten



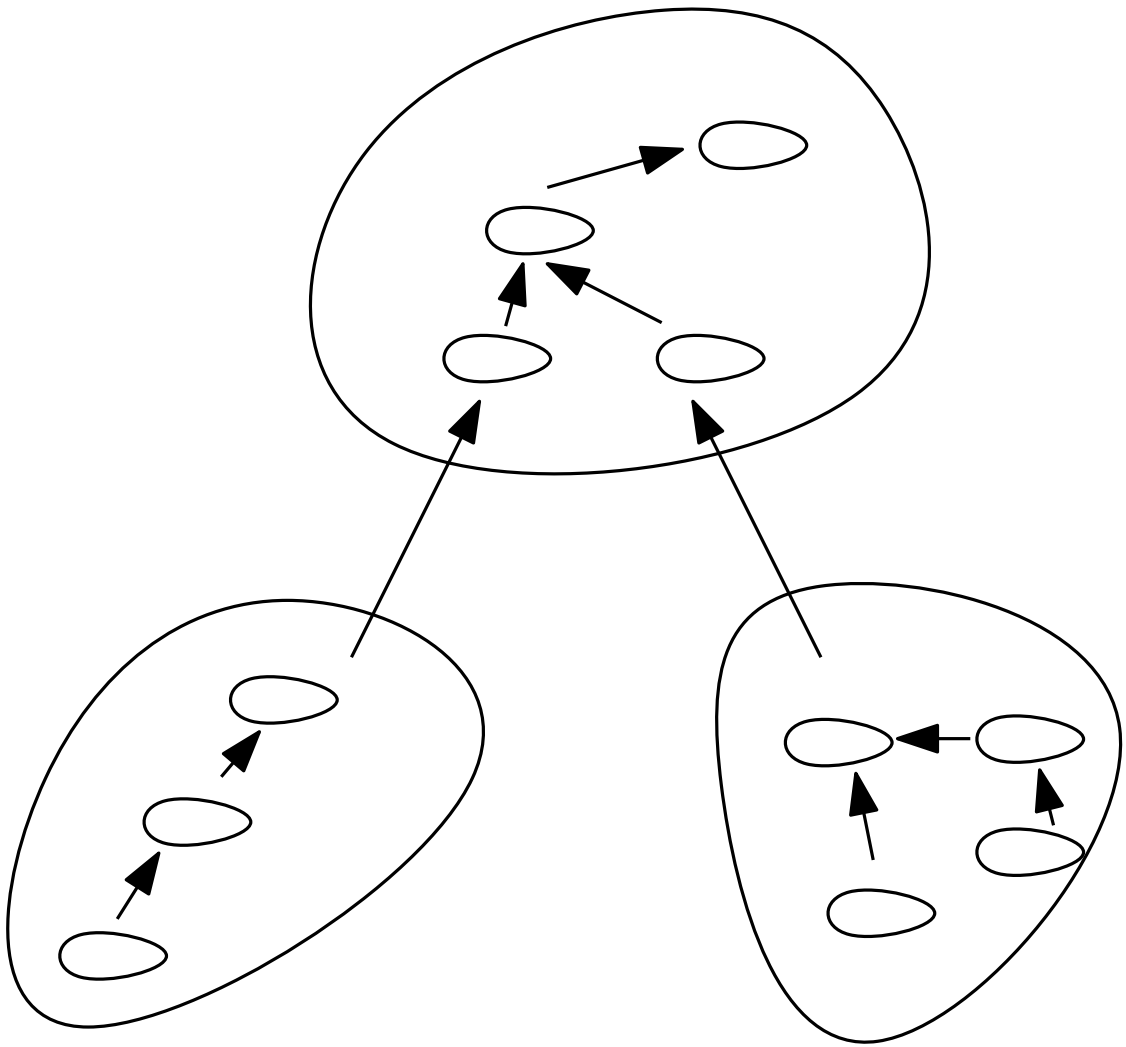
Metaclustergröße=$\Theta(\ell)$

„abhängiges“ randomisiertes Runden garantiert, dass Metaclustervolumen höchstens auf- oder abgerundet wird

Kapazitätsverletzung $\approx \frac{(2-\delta)u}{\lfloor 2-\delta \rfloor u} + \frac{1}{\ell}$

Erwartete Kosten durch Umleitung in benachbartes Metacluster $O(\ell^2 \cdot d_{\mathsf{av}}(j))$

Hierarchische Metacluster – Kapazitäten



Metaclustergröße=$\Theta(\ell)$

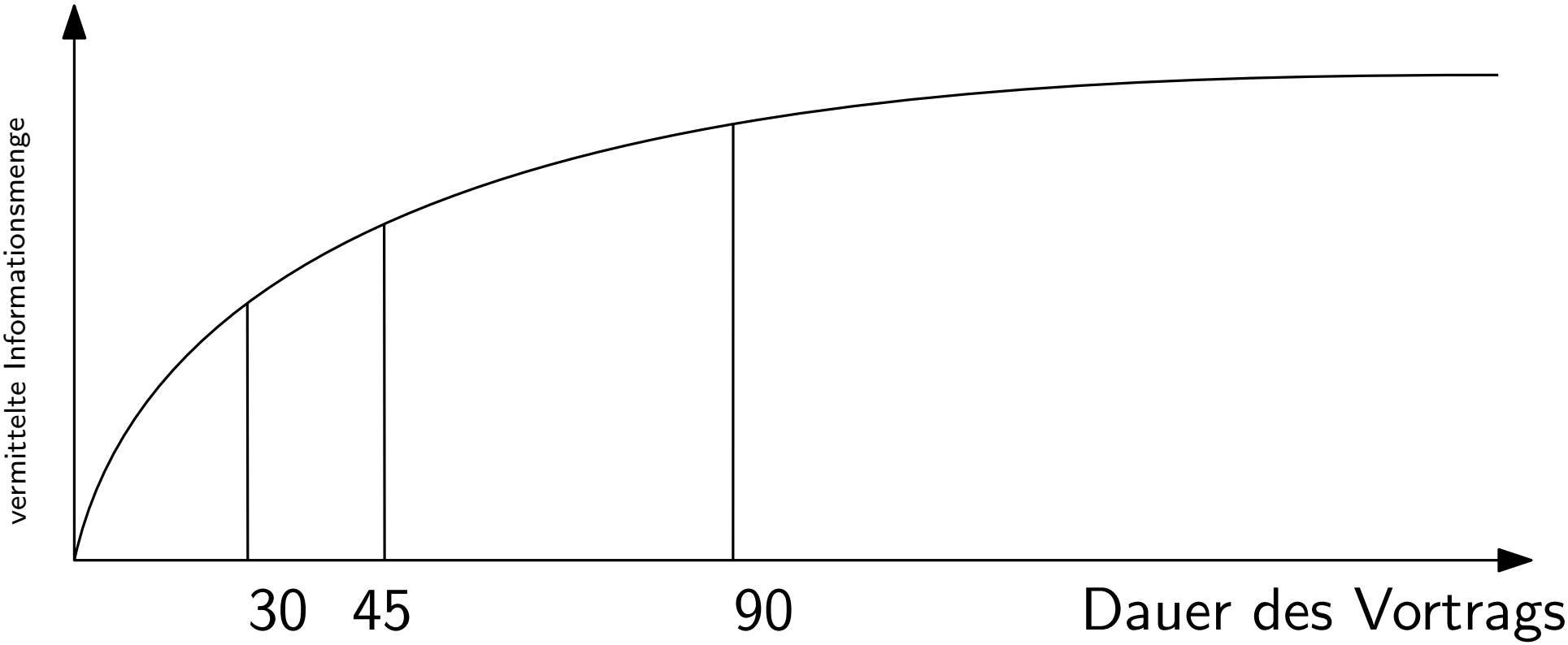
„abhängiges“ randomisiertes Runden garantiert, dass Metaclustervolumen höchstens auf- oder abgerundet wird

Kapazitätsverletzung $\approx \frac{(2-\delta)u}{\lfloor 2-\delta \rfloor u} + \frac{1}{\ell}$

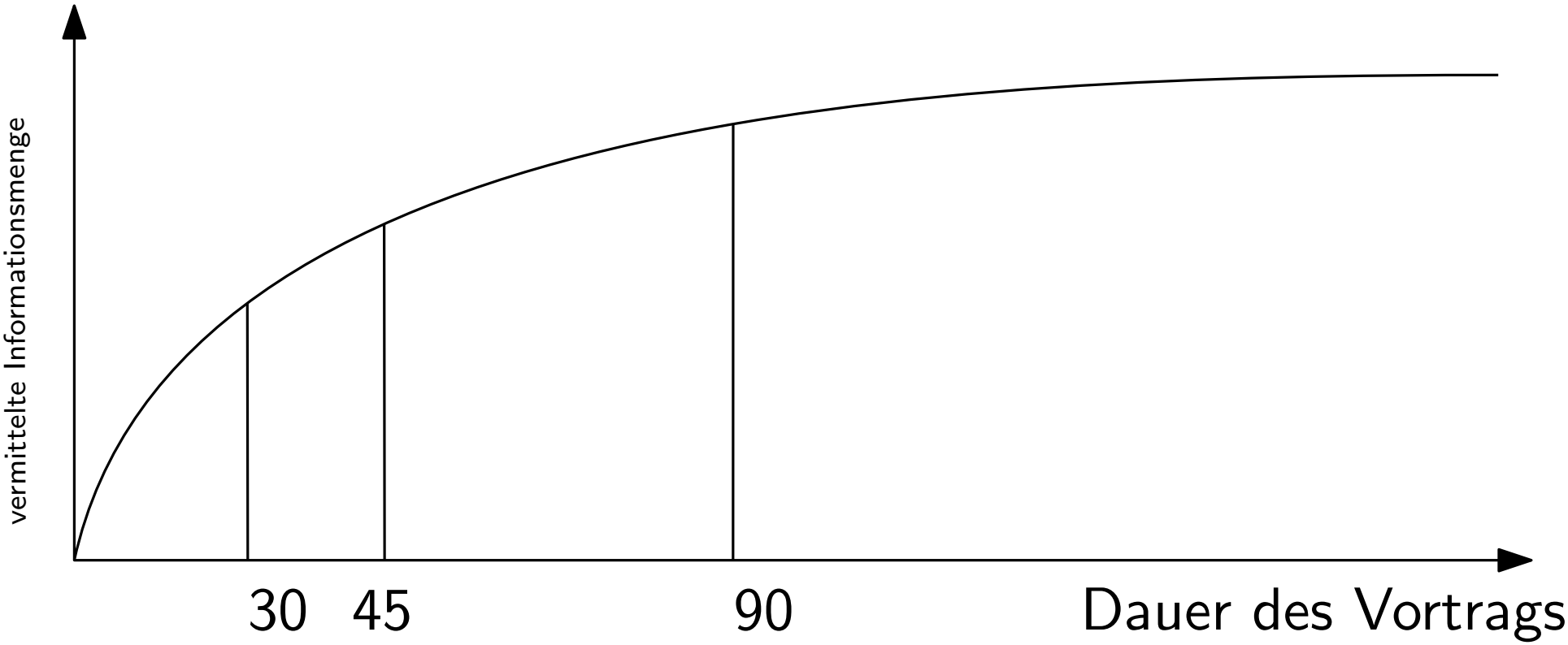
Erwartete Kosten durch Umleitung in benachbartes Metacluster $O(\ell^2 \cdot d_{\mathsf{av}}(j))$

Bifaktor $(2+\epsilon, O(1/\epsilon^2))$

Submodulare Funktionen

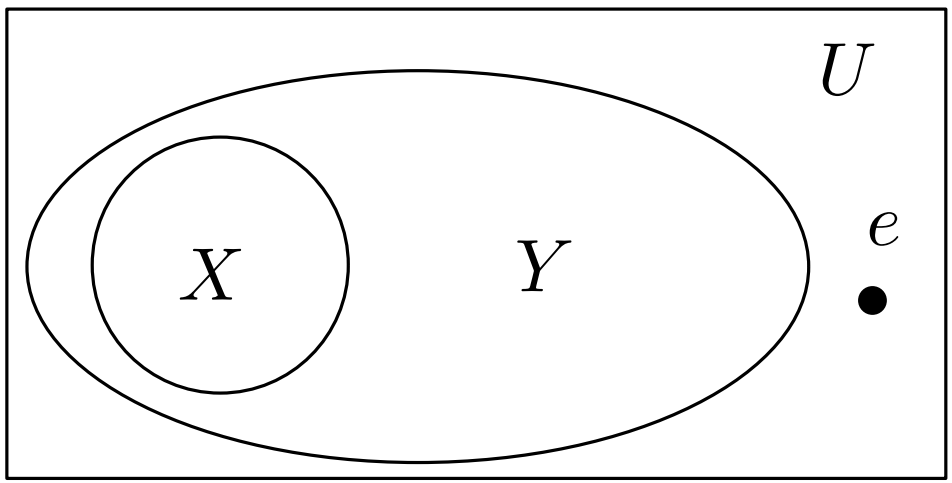


Submodulare Funktionen

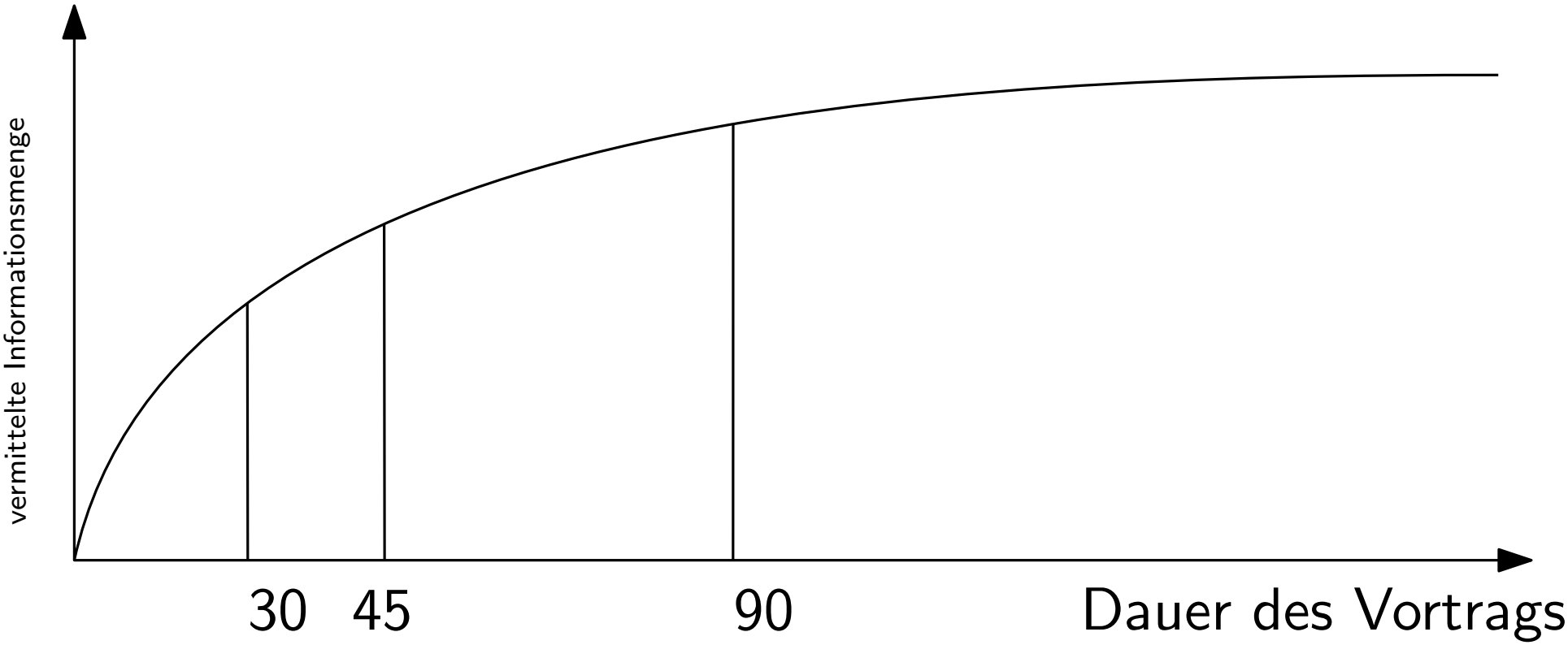


$$f(X \cup \{e\}) - f(X) \geq f(Y \cup \{e\}) - f(Y)$$ Submodularität

$$\forall X \subseteq Y \subseteq U, e \in U - Y$$



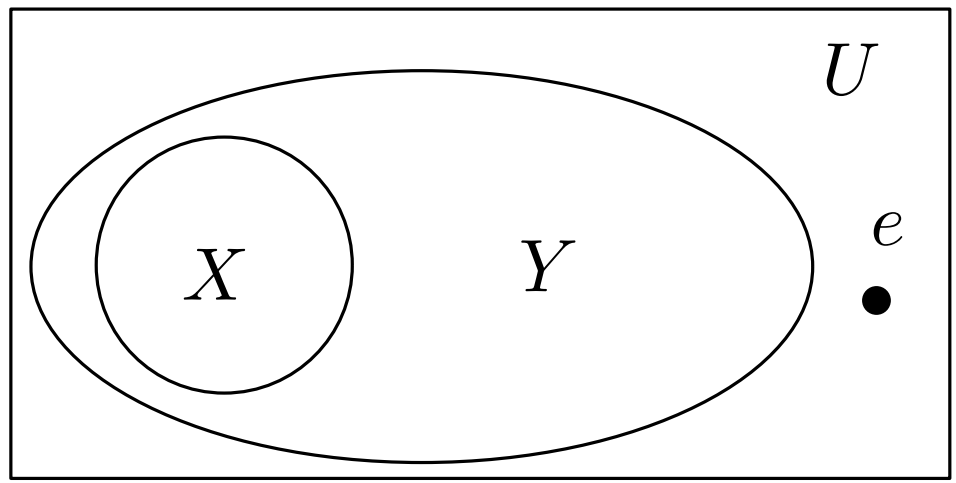
Submodulare Funktionen



$$f(X \cup \{e\}) - f(X) \geq f(Y \cup \{e\}) - f(Y) \quad \text{Submodularität}$$

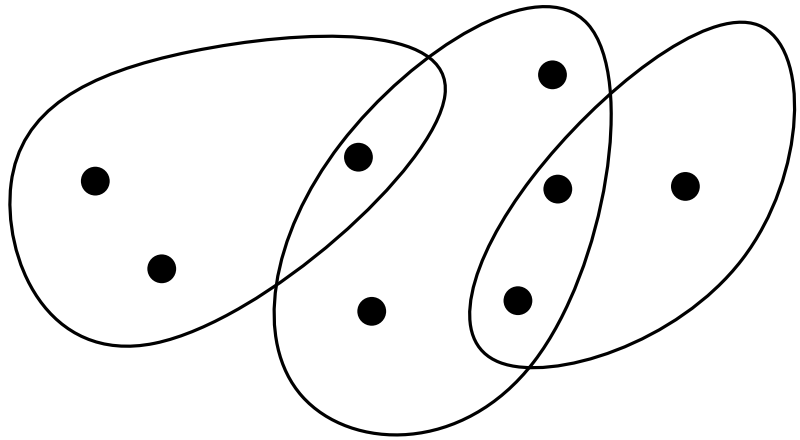
$$f(X) \leq f(Y) \quad \text{Monotonie}$$

$$\forall X \subseteq Y \subseteq U, e \in U - Y$$

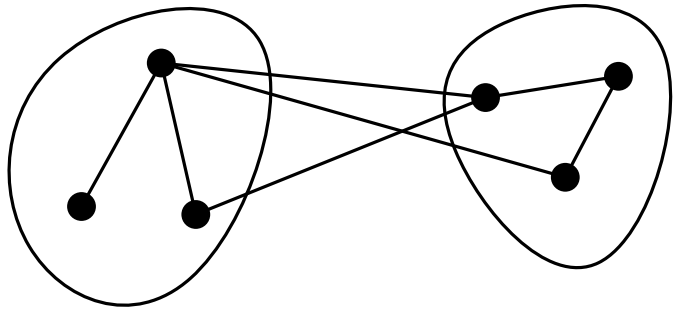


Beispiele Submodularer Funktionen

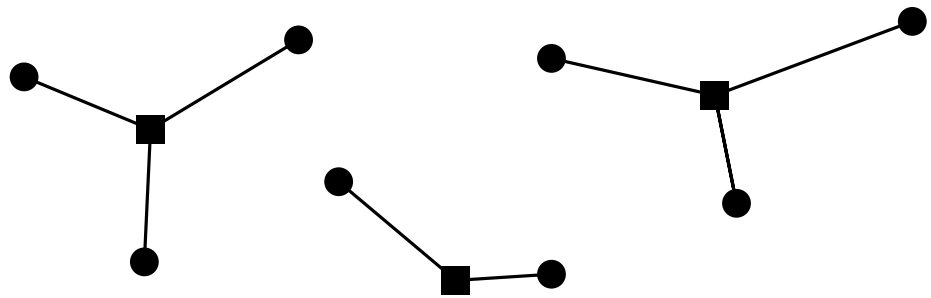
MaxCoverage



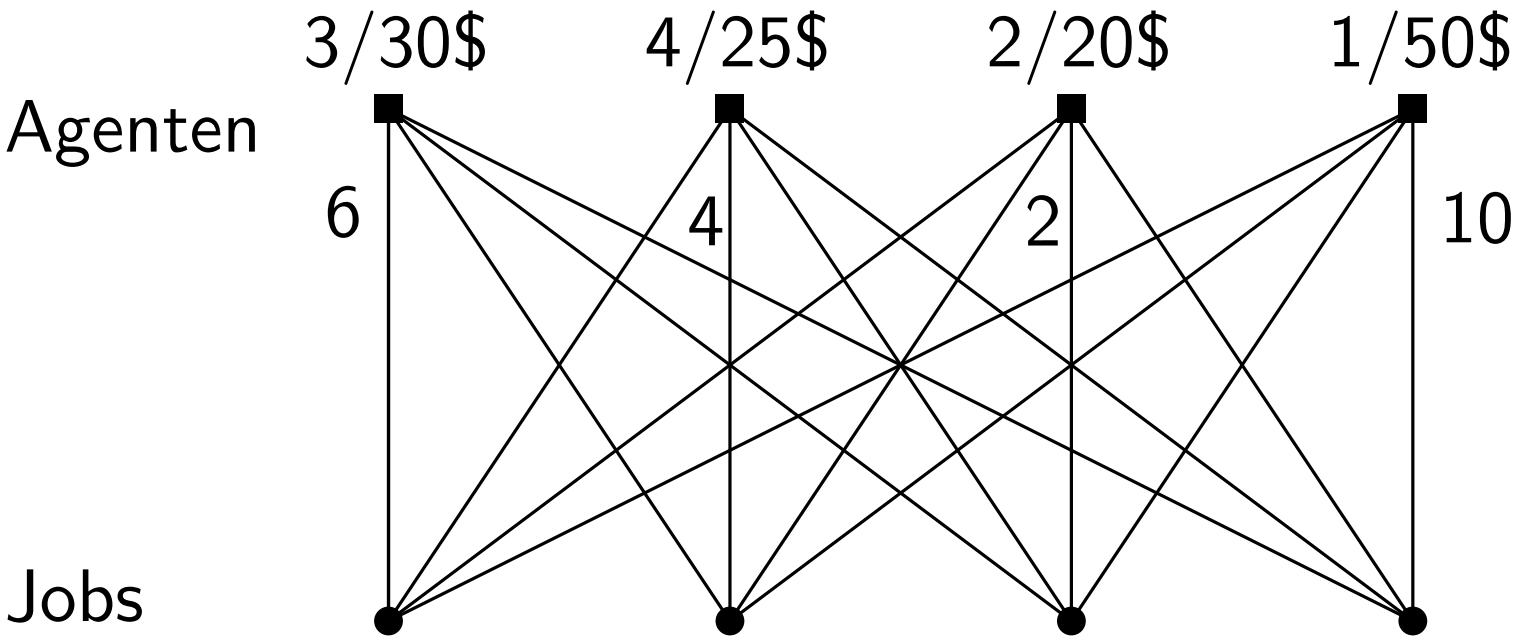
MaxCut



MaxFacilityLocation



Packungs- und Überdeckungsconstraints



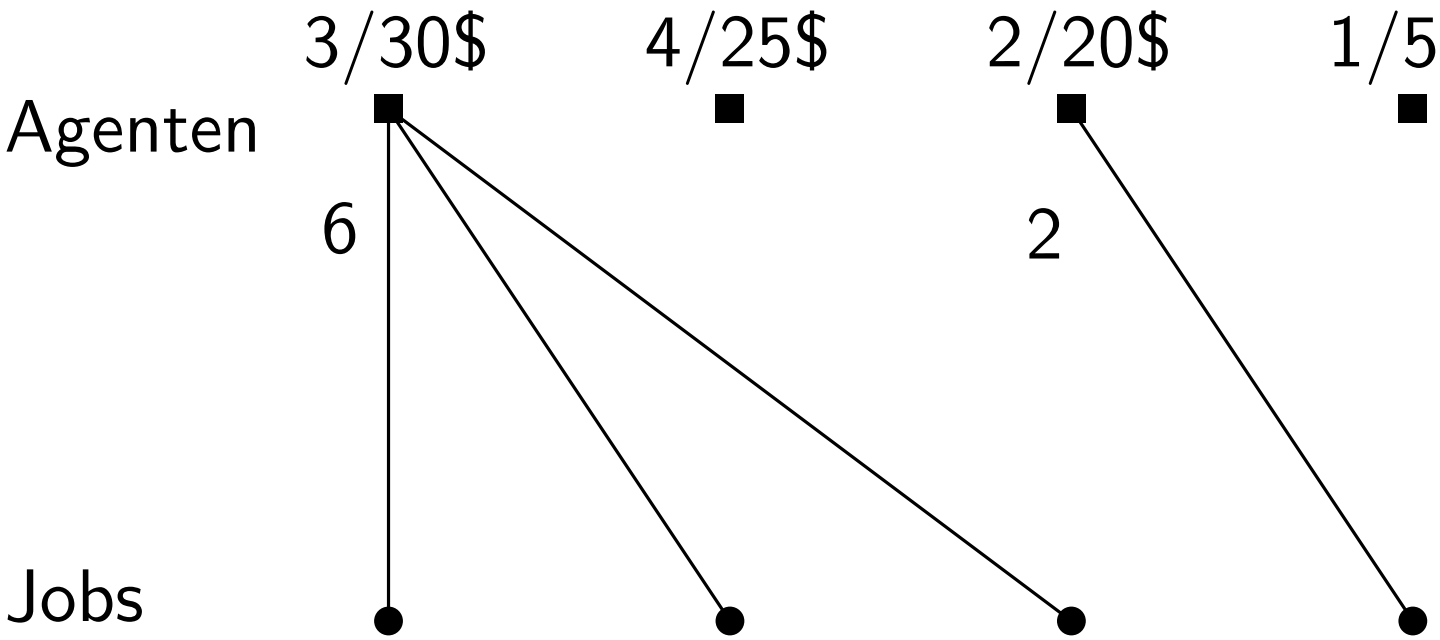
$W = 50\$$

Gegeben: monotone, submodulare Funktion $f : 2^U \to \mathbb{Q}^+$, für jedes $e \in U$ ein Profit p_e und ein Gewicht w_e, eine Profitanforderung P und ein Budget W

Gesucht: Menge $X \subseteq U$, mit $f(X)$ maximal, so dass $\sum_{e \in X} p_e \geq P$ und $\sum_{e \in X} w_e \leq W$

gemeinsame Arbeit mit Sumedha Uniyal (submitted '17)

Packungs- und Überdeckungsconstraints

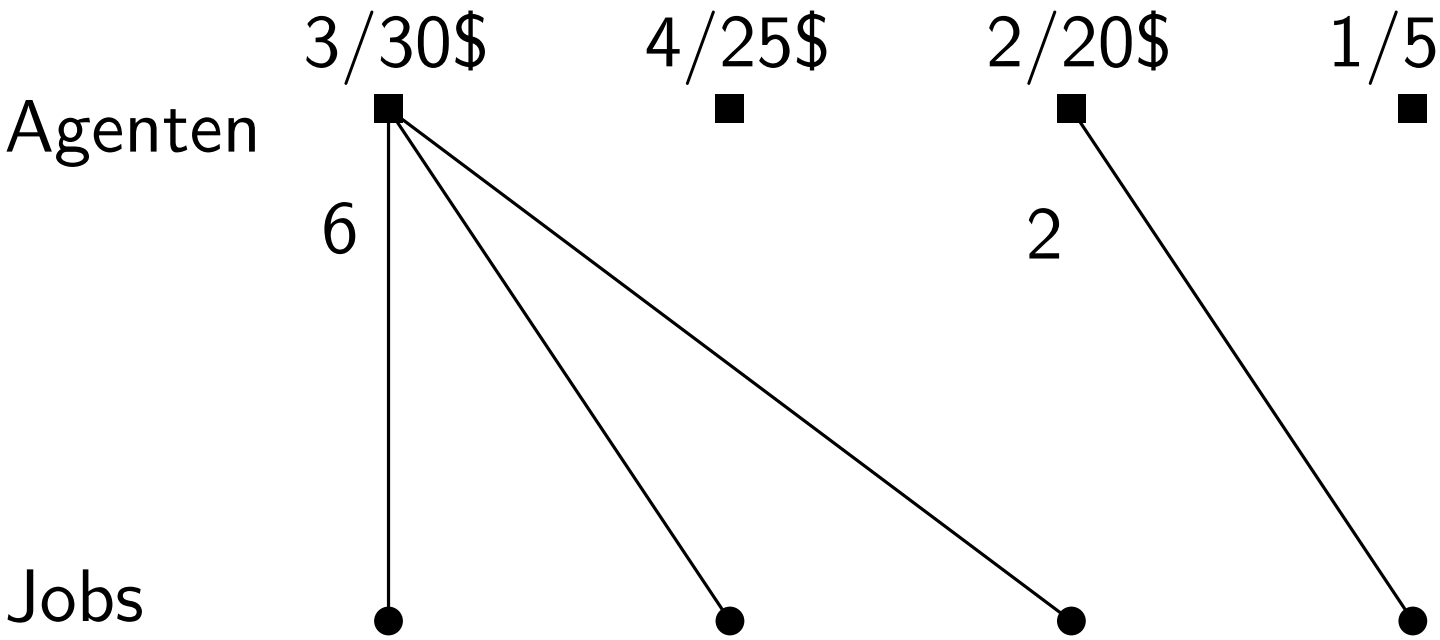


$W = 50\$$

Gegeben: monotone, submodulare Funktion $f : 2^U \to \mathbb{Q}^+$, für jedes $e \in U$ ein Profit p_e und ein Gewicht w_e, eine Profitanforderung P und ein Budget W

Gesucht: Menge $X \subseteq U$, mit $f(X)$ maximal, so dass $\sum_{e \in X} p_e \geq P$ und $\sum_{e \in X} w_e \leq W$

Packungs- und Überdeckungsconstraints



$W = 50\$$

Gegeben: monotone, submodulare Funktion $f : 2^U \to \mathbb{Q}^+$, für jedes $e \in U$ ein Profit p_e und ein Gewicht w_e, eine Profitanforderung P und ein Budget W

Gesucht: Menge $X \subseteq U$, mit $f(X)$ maximal, so dass $\sum_{e \in X} p_e \geq P$ und $\sum_{e \in X} w_e \leq W$

schwach NP-vollst.
(Subset-Sum)

Stand der Forschung

- Kardinalitätsconstraint: $1 - 1/e$ via Greedy [Nemhauser et al., Math. Progr. 78]

Stand der Forschung

- Kardinalitätsconstraint: $1 - 1/e$ via Greedy [Nemhauser et al., Math. Progr. 78]
- Faktor $1 - 1/e$ ist scharf [Feige, STOC'96]

Stand der Forschung

- Kardinalitätsconstraint: $1 - 1/e$ via Greedy [Nemhauser et al., Math. Progr. 78]
- Faktor $1 - 1/e$ ist scharf [Feige, STOC'96]
- Packungsconstraint: $1 - 1/e$ via Greedy [Sviridenko, ORL'04]

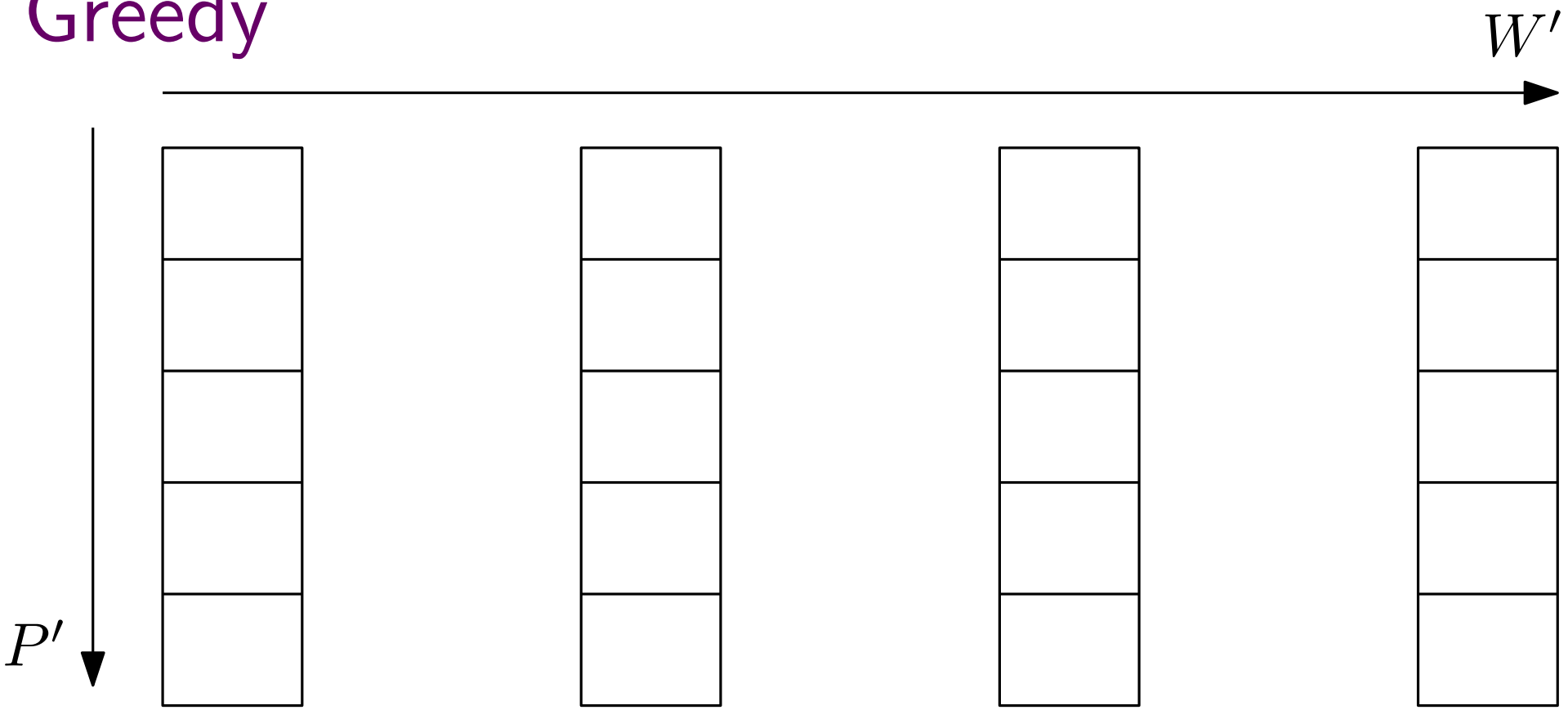
Stand der Forschung

- Kardinalitätsconstraint: $1 - 1/e$ via Greedy [Nemhauser et al., Math. Progr. 78]
- Faktor $1 - 1/e$ ist scharf [Feige, STOC'96]
- Packungsconstraint: $1 - 1/e$ via Greedy [Sviridenko, ORL'04]
- $O(1)$ Packungsconstraints: $1 - 1/e - \epsilon$ via Multilinearer Relaxierung [Kulik et al., SODA'09]

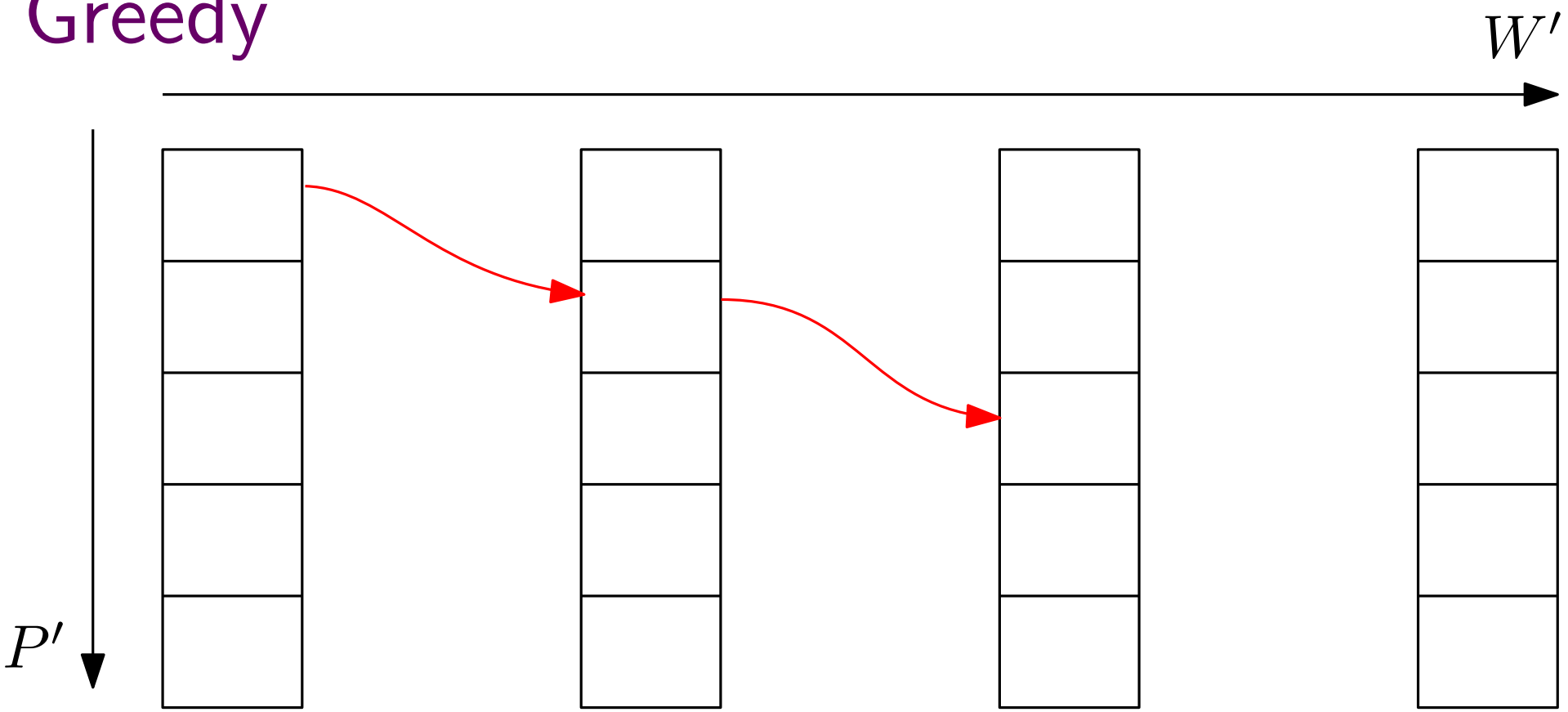
Stand der Forschung

- Kardinalitätsconstraint: $1 - 1/e$ via Greedy [Nemhauser et al., Math. Progr. 78]
- Faktor $1 - 1/e$ ist scharf [Feige, STOC'96]
- Packungsconstraint: $1 - 1/e$ via Greedy [Sviridenko, ORL'04]
- $O(1)$ Packungsconstraints: $1 - 1/e - \epsilon$ via Multilinearer Relaxierung [Kulik et al., SODA'09]
- Matroid Constraint: 0.309 via multilinearer Relaxierung [Vondrak, STOC'09]

Greedy

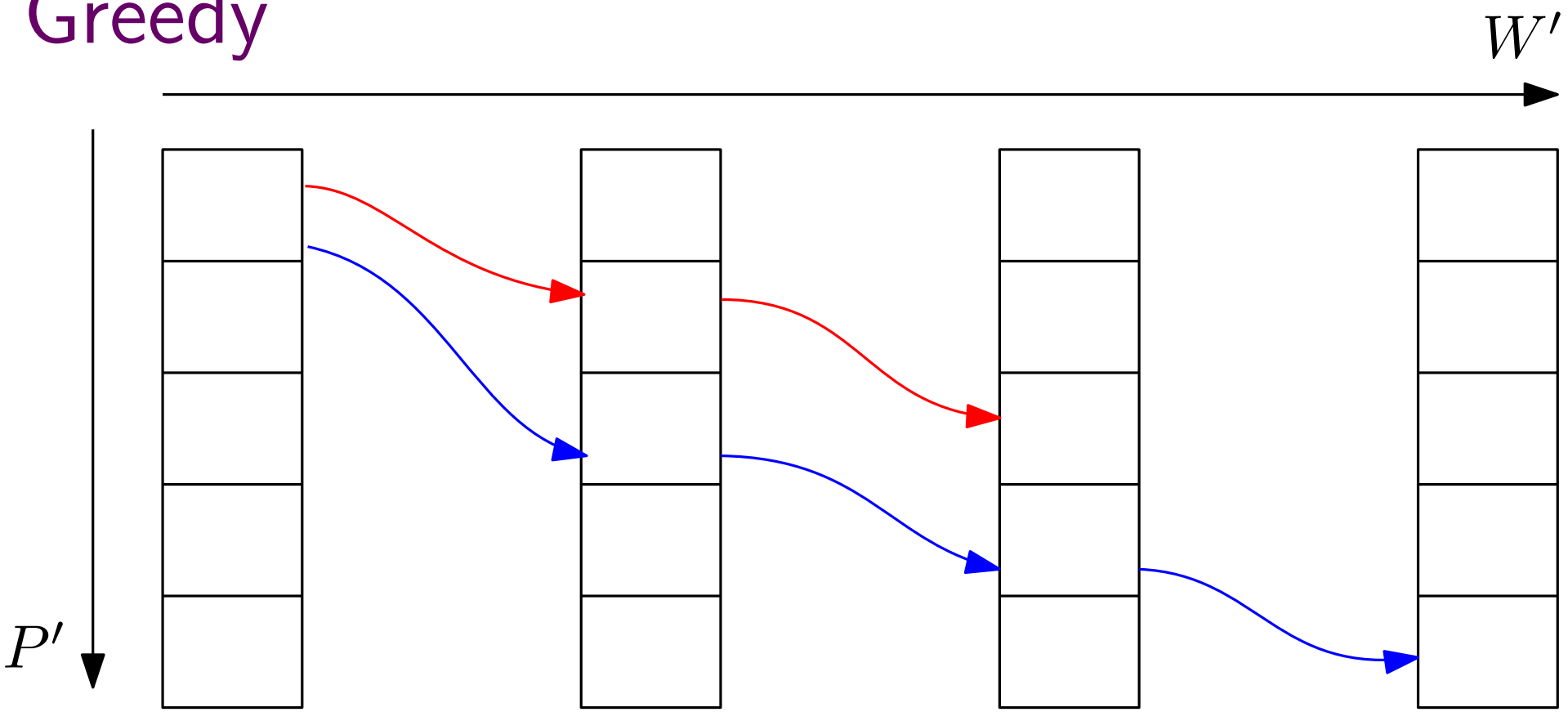


Greedy



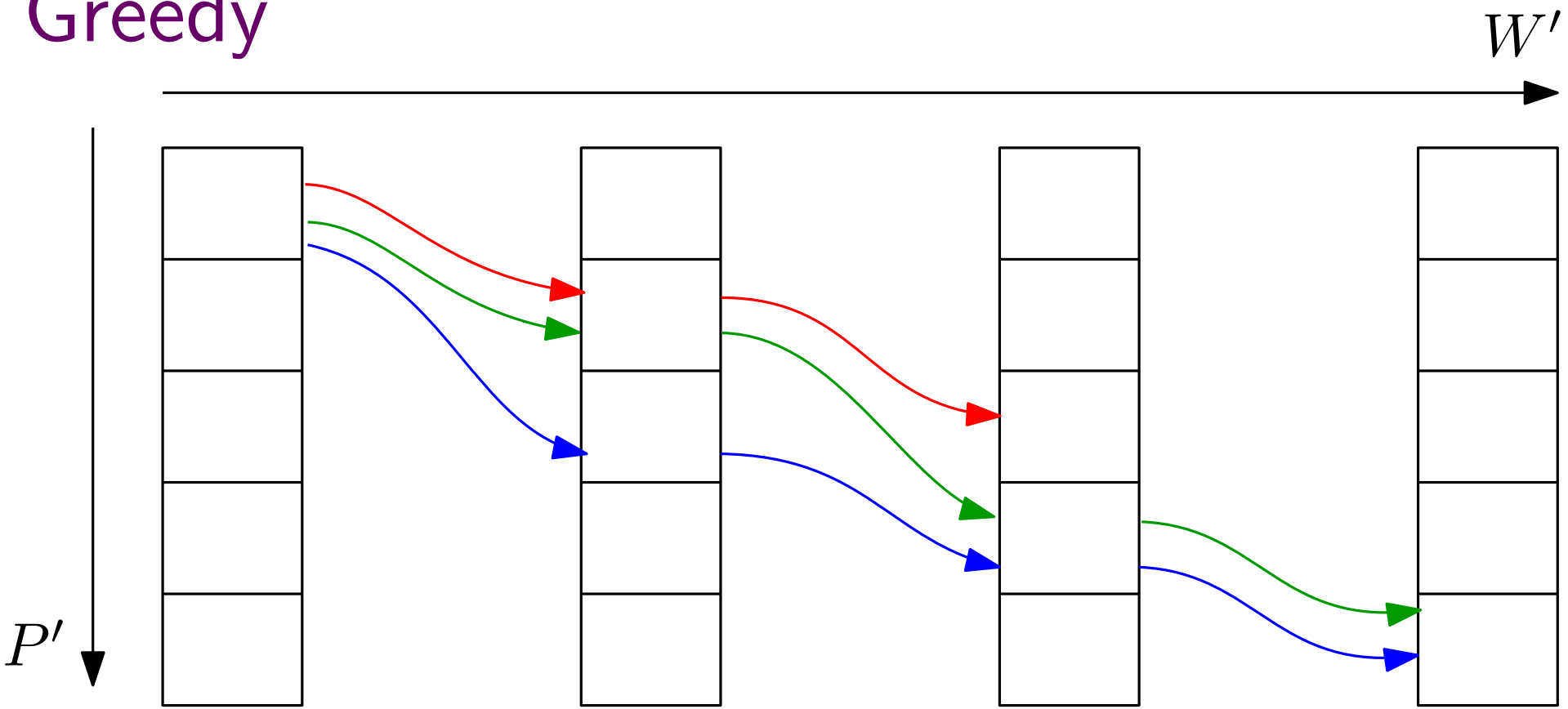
greedy kann in „Sackgasse“ geraten

Greedy



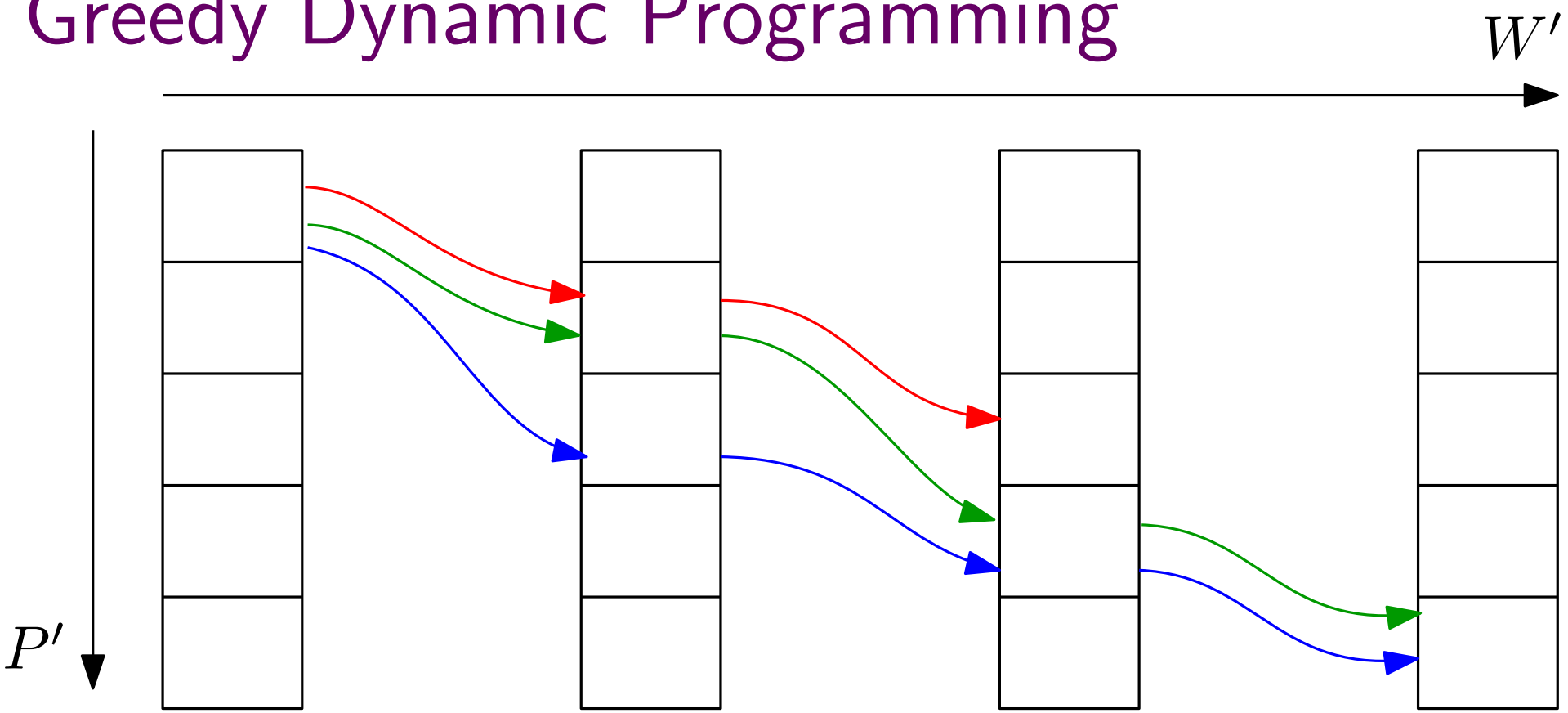
Gewichte/Profit wie in optimaler Lösung

Greedy



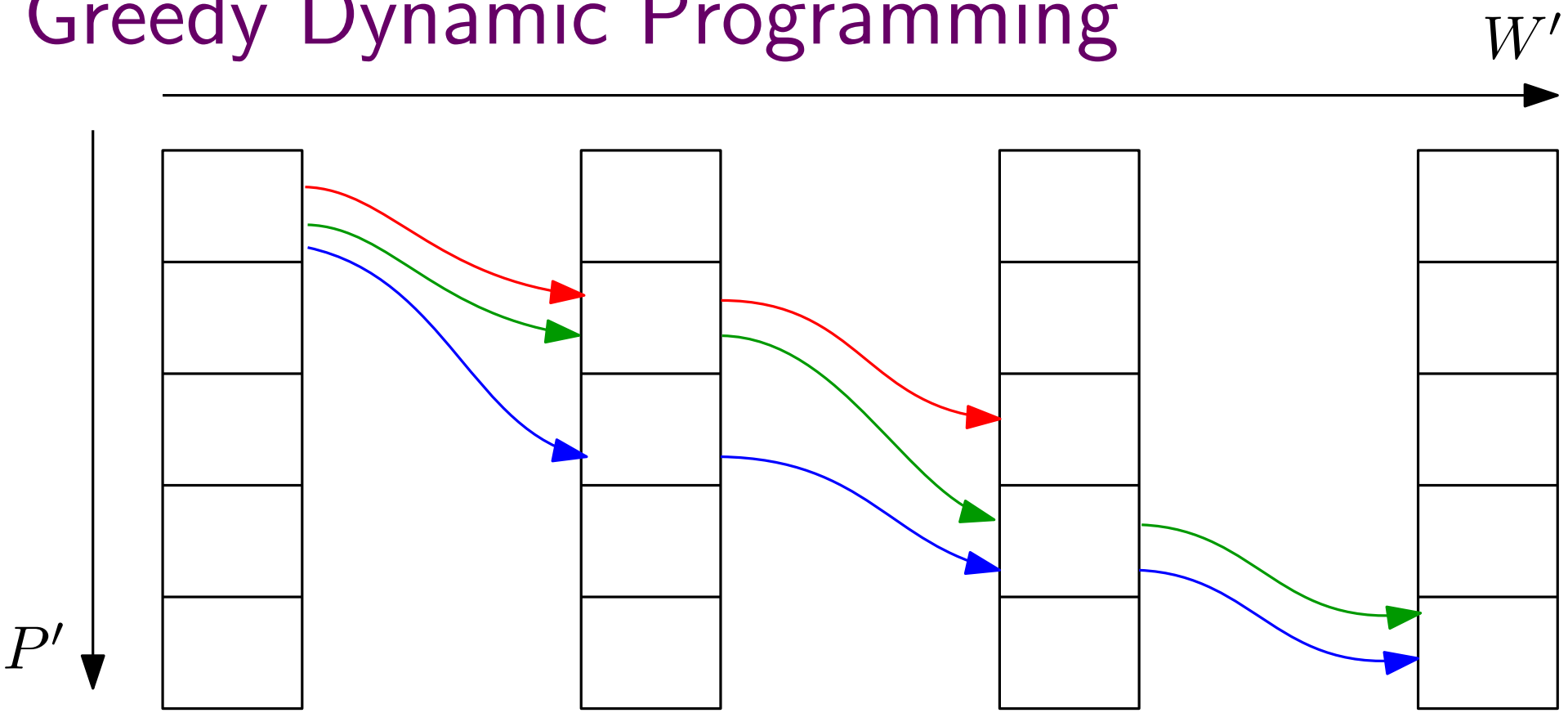
beste Greedy-Sequenz

Greedy Dynamic Programming



Tabelleneintrag $T[l, W', P']$ speichert l-elementige,
approximative Lösung mit Profit P' und Gewicht W'

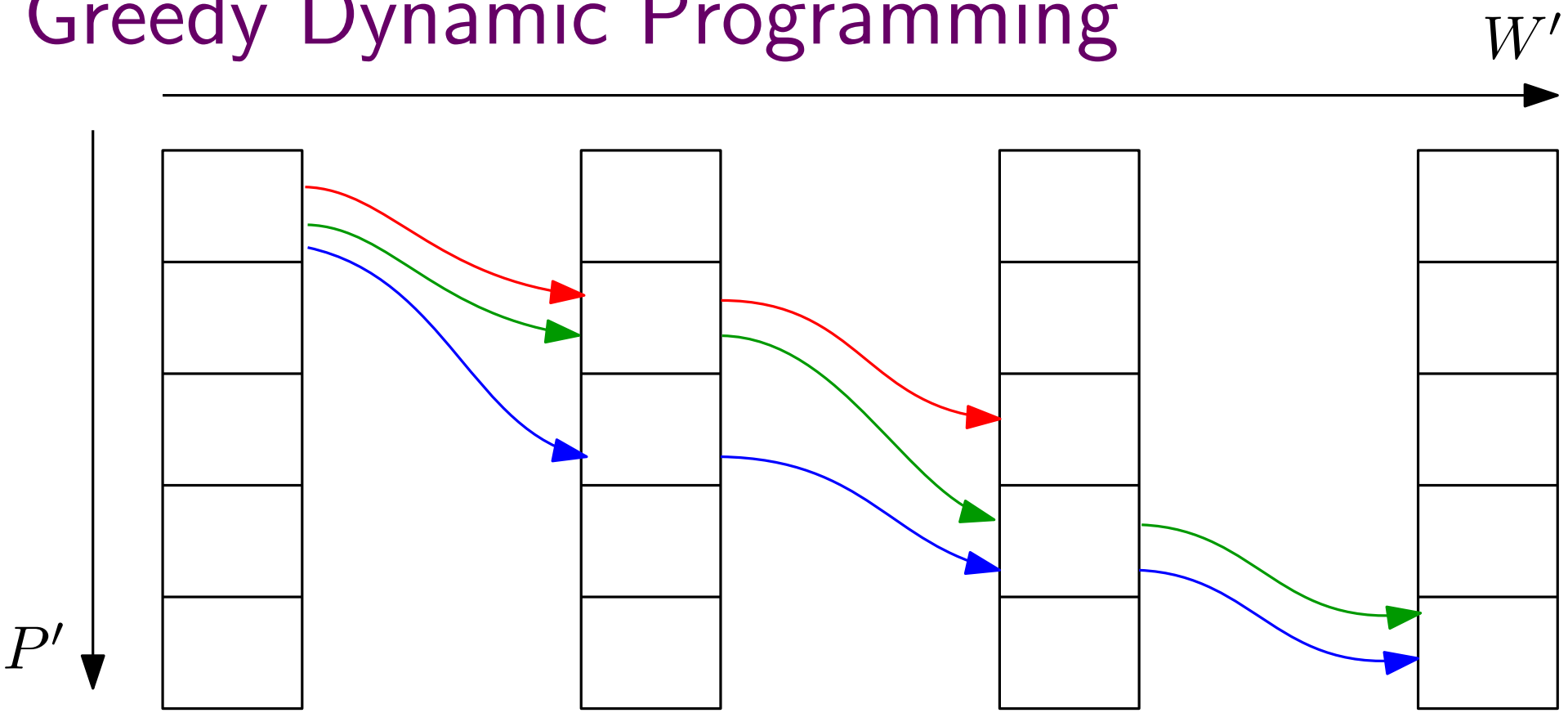
Greedy Dynamic Programming



Tabelleneintrag $T[l, W', P']$ speichert l-elementige, *approximative* Lösung mit Profit P' und Gewicht W'

um $T[l, W', P']$ zu berechnen wähle beste Erweiterung eines Eintrags $T[l-1, W'', P'']$ um ein Element in „zulässiger“ Weise

Greedy Dynamic Programming

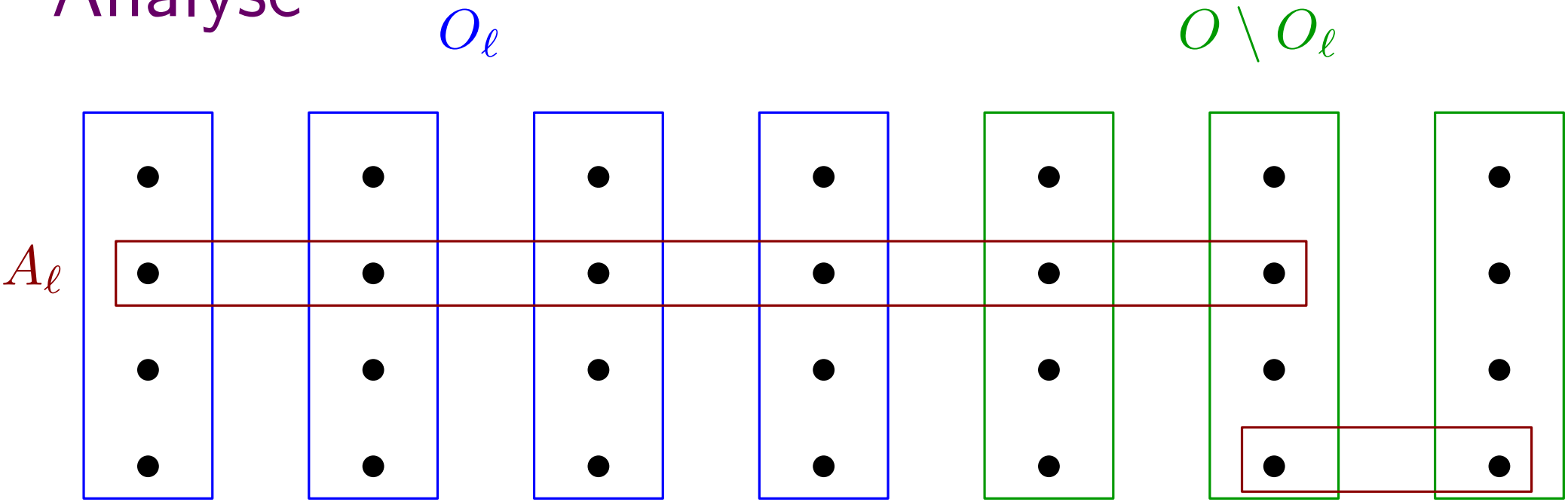


Tabelleneintrag $T[l, W', P']$ speichert l-elementige, *approximative* Lösung mit Profit P' und Gewicht W'

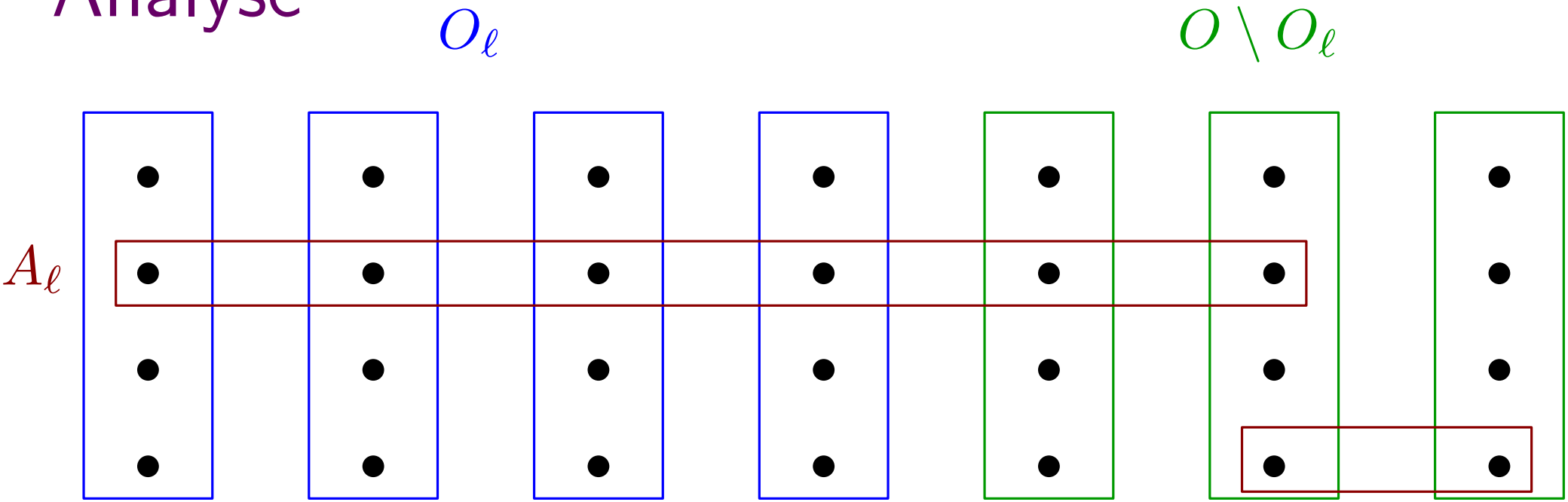
um $T[l, W', P']$ zu berechnen wähle beste Erweiterung eines Eintrags $T[l-1, W'', P'']$ um ein Element in „zulässiger“ Weise

gib beste Lösung $T[l, P', W']$ mit $P' \geq P/2$ und $W' \leq W$ aus

Analyse

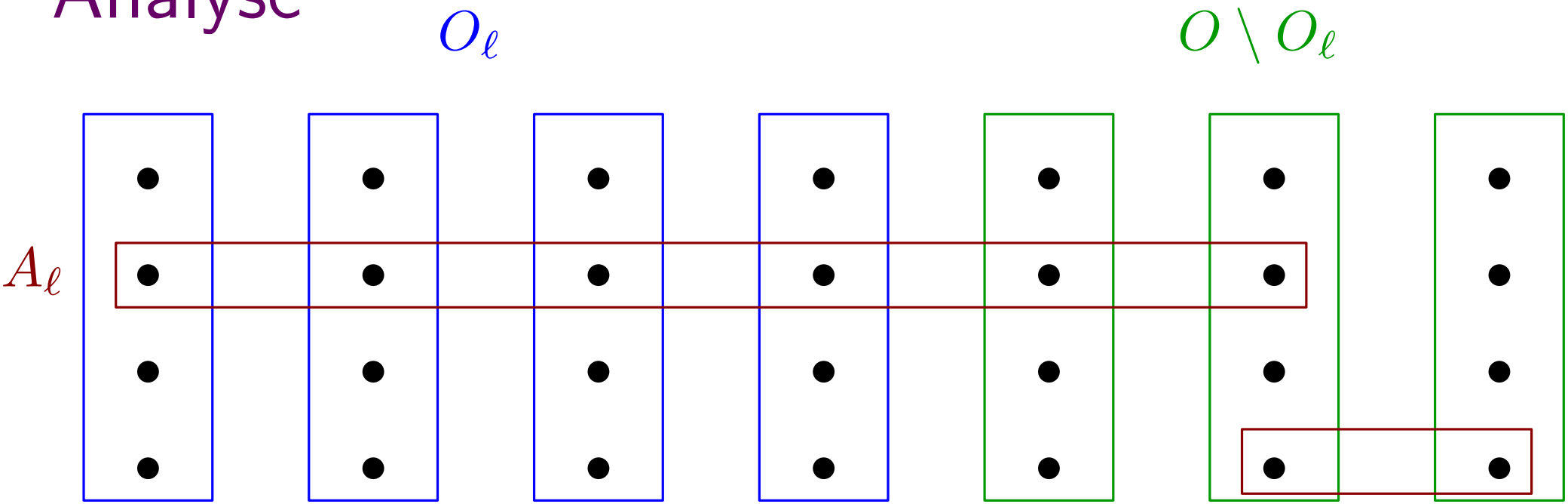


Analyse



Invariante: $f(A_\ell) \geq \frac{1}{2} g(O_\ell)$

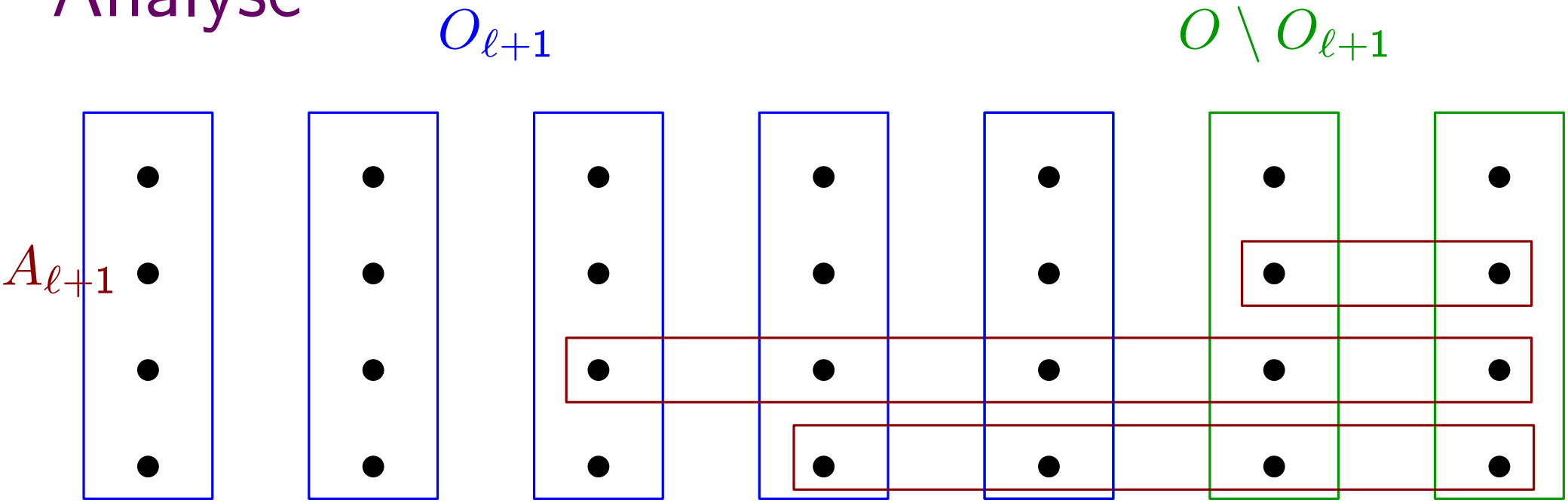
Analyse



Invariante: $f(A_\ell) \geq \frac{1}{2} g(O_\ell)$

erweitere solange $e \in O \setminus O_\ell$ mit $f(A_\ell \cup \{e\}) - f(A_\ell) > \frac{1}{2} g(e)$ existiert, erweitere O_ℓ um e zu $O_{\ell+1}$

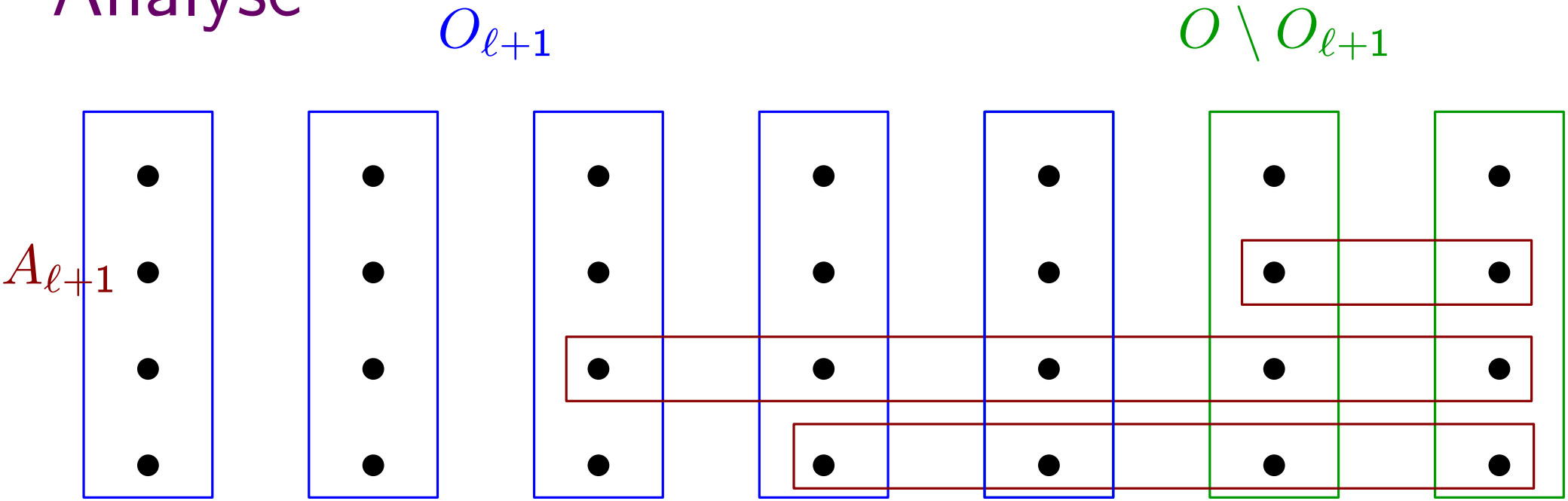
Analyse



Invariante: $f(A_\ell) \geq \frac{1}{2} g(O_\ell)$

erweitere solange $e \in O \setminus O_\ell$ mit $f(A_\ell \cup \{e\}) - f(A_\ell) > \frac{1}{2} g(e)$
existiert, erweitere O_ℓ um e zu $O_{\ell+1}$

Analyse



Invariante: $f(A_\ell) \geq \frac{1}{2} g(O_\ell)$

erweitere solange $e \in O \setminus O_\ell$ mit $f(A_\ell \cup \{e\}) - f(A_\ell) > \frac{1}{2} g(e)$ existiert, erweitere O_ℓ um e zu $O_{\ell+1}$

Falls kein solches e existiert, gilt zusätzlich $f(A_\ell) \geq \frac{1}{2} g(O \setminus O_\ell)$

Analyse

Zwei-Phasen-Ansatz

Am Ende der ersten Phase gilt $f(A_\ell) \geq \frac{1}{2} g(O_\ell)$ und $f(A_\ell) \geq \frac{1}{2} g(O \setminus O_\ell)$

Analyse

Zwei-Phasen-Ansatz

Am Ende der ersten Phase gilt $f(A_\ell) \geq \frac{1}{2} g(O_\ell)$ und $f(A_\ell) \geq \frac{1}{2} g(O \setminus O_\ell)$

Schlechteste Verteilung: $g(O_\ell) = g(O \setminus O_\ell) = \frac{1}{2}\mathsf{OPT}$.

Analyse

Zwei-Phasen-Ansatz

Am Ende der ersten Phase gilt $f(A_\ell) \geq \frac{1}{2} g(O_\ell)$ und $f(A_\ell) \geq \frac{1}{2} g(O \setminus O_\ell)$

Schlechteste Verteilung: $g(O_\ell) = g(O \setminus O_\ell) = \frac{1}{2}\mathsf{OPT}$.

$\rightsquigarrow$ Approximationsfaktor $1/4$

Analyse

Zwei-Phasen-Ansatz

Am Ende der ersten Phase gilt $f(A_\ell) \geq \frac{1}{2} g(O_\ell)$ und $f(A_\ell) \geq \frac{1}{2} g(O \setminus O_\ell)$

Schlechteste Verteilung: $g(O_\ell) = g(O \setminus O_\ell) = \frac{1}{2}\mathsf{OPT}$.

$\rightsquigarrow$ Approximationsfaktor $1/4$

Verallgemeinerung auf m Phasen:

Analyse

Zwei-Phasen-Ansatz

Am Ende der ersten Phase gilt $f(A_\ell) \geq \frac{1}{2} g(O_\ell)$ und $f(A_\ell) \geq \frac{1}{2} g(O \setminus O_\ell)$

Schlechteste Verteilung: $g(O_\ell) = g(O \setminus O_\ell) = \frac{1}{2}\mathsf{OPT}$.

$\rightsquigarrow$ Approximationsfaktor $1/4$

Verallgemeinerung auf m Phasen:

am Ende der i-ten Phase gilt:

$$f(A_{\ell_i} \cup A_{\ell_{i-1}}) - f(A_{\ell_i}) \geq \left(1 - \tfrac{i}{m}\right) g(O_{\ell_i} \setminus O_{\ell_{i-1}})$$

$$f(A_{\ell_i}) \geq \tfrac{i}{m} g(O \setminus O_{\ell_i})$$

Factor-Revealing LP

ungünstigste Verteilung auf die Phasen lässt sich schwerer raten...

Factor-Revealing LP

ungünstigste Verteilung auf die Phasen lässt sich schwerer raten...

... aber mittels eines LPs ermitteln:

$$\begin{aligned}
&\min a_m && \text{s.t.} && \\
&\quad a_1 \geq \left(1 - \tfrac{1}{m}\right) o_1; && && (1)\\
&\quad a_i \geq a_{i-1} + \left(1 - \tfrac{i}{m}\right) o_i && \forall i \in [m] \setminus \{1\}; && (2)\\
&\quad a_i \geq \tfrac{i}{m} \left(1 - \textstyle\sum_{j \leq i} o_j\right) && \forall i \in [m]; && (3)\\
&\quad a_i \geq 0,\ o_i \geq 0 && \forall i \in [m]. && (4)
\end{aligned}$$

Wert des LPs ist untere Schranke für Approximationsgüte

konvergiert gegen $1/e$ für $m \to \infty$

Reduktion der Constraint-Verletzung auf $1 + \epsilon$

Sequenz stagniert, wenn $O \setminus O_\ell \subseteq A_\ell$

Reduktion der Constraint-Verletzung auf $1 + \epsilon$

Sequenz stagniert, wenn $O \setminus O_\ell \subseteq A_\ell$

reserviere minimale „Fluchtmengen“ $F_{W'}$, welche die gemäß p_e/w_e „effizientesten“ Elemente mit Gesamtgewicht $\geq W - W'$ enthalten

Reduktion der Constraint-Verletzung auf $1 + \epsilon$

Sequenz stagniert, wenn $O \setminus O_\ell \subseteq A_\ell$

reserviere minimale „Fluchtmengen“ $F_{W'}$, welche die gemäß p_e/w_e „effizientesten“ Elemente mit Gesamtgewicht $\geq W - W'$ enthalten

garantiert, dass sich eine stagnierte Sequenz mittels $F_{W'}$ zu zulässiger Lösung komplettieren lässt

Reduktion der Constraint-Verletzung auf $1 + \epsilon$

Sequenz stagniert, wenn $O \setminus O_\ell \subseteq A_\ell$

reserviere minimale „Fluchtmengen“ $F_{W'}$, welche die gemäß p_e/w_e „effizientesten“ Elemente mit Gesamtgewicht $\geq W - W'$ enthalten

garantiert, dass sich eine stagnierte Sequenz mittels $F_{W'}$ zu zulässiger Lösung komplettieren lässt

Schwierigkeit bei der Analyse: $F_{W'}$ könnte Elemente aus O enthalten

Erweitertes Factor-Revealing LP

min c s.t.

$$a_0 = o_0; \tag{5}$$

$$a_i \geq a_{i-1} + \left(1 - \tfrac{i}{m}\right) o_i \quad \forall i \in [m]; \tag{6}$$

$$b_i \geq a_i + g_i \quad \forall i \in [m] \cup \{0\}; \tag{7}$$

$$a_i \geq \tfrac{i}{m}(1 - f_i - \textstyle\sum_{j \leq i} o_j) + f_i - g_i \quad \forall i \in [m] \cup \{0\}; \tag{8}$$

$$b_i \geq f_i \quad \forall i \in [m] \cup \{0\}; \tag{9}$$

$$f_i \leq f_{i-1} \quad \forall i \in [m]; \tag{10}$$

$$g_i \leq f_i \quad \forall i \in [m] \cup \{0\}; \tag{11}$$

$$f_j + \textstyle\sum_{i=0}^{j} o_i \leq 1; \quad \forall j \in [m] \cup \{0\}; \tag{12}$$

$$c \geq b_i \quad \forall i \in [m] \cup \{0\}; \tag{13}$$

$$a_i \geq 0,\ o_i \geq 0\ f_i \geq 0\ g_i \geq 0 \quad \forall i \in [m] \cup \{0\}. \tag{14}$$

Grenzwert ist mindestens 0.353 aber echt kleiner als $1/e \approx 0.368$

Das Resultat

Satz Für jedes $\epsilon > 0$ gibt einen 0.353-Approximationsalgorithmus zur Maximierung einer submodularen Funktion unter einem Packungs- und einem Überdeckungsconstraint, der beide Constraints um einen Faktor $1 + \epsilon$ verletzt und eine Laufzeit von $n^{O(1/\epsilon)}$ besitzt

Das Resultat

Satz Für jedes $\epsilon > 0$ gibt einen 0.353-Approximationsalgorithmus zur Maximierung einer submodularen Funktion unter einem Packungs- und einem Überdeckungsconstraint, der beide Constraints um einen Faktor $1 + \epsilon$ verletzt und eine Laufzeit von $n^{O(1/\epsilon)}$ besitzt

Kor. Es gibt einen 2.29-Approximationsalgorithmus für k-Median mit harten, non-uniformen Kapazitäten, wenn die zugrundeliegende Metrik nur zwei verschiedene Abstände zwischen Clients und Facilities besitzt. Die bestmögliche Approximation (außer P=NP) ist 1.735

Weitere Ergebnisse der Arbeit

Standortprobleme

- verbesserte $O(1)$-Approximation für das Knapsack-Median-Problem [ESA'15]
- $(1 - 1/e)$-Approximation für Maximum-Betweenness-Centrality [WALCOM'11]
- PTAS für geometrische Coverage-Probleme

Netzwerkdesign

- Maximum-Edge-Disjoint-Paths [ESA'16]
- Netzwerkdesign mit beschränkten Distanzen [STACS'15]
- Gradbasierte Spannbaumprobleme [Algorithmica'17, ToCS'15]
- geometrisches ND: Box-Repräsentationen [ESA'14], Manhattan-Networks [ESA'11, ISAAC'13], Non-Crossing-Steinerforests [ISAAC'15],