Computing the nucleolus of cooperative games

Lexicographical optimisation with LPs

Kempten Autumn Talks
11 November 2020

Márton Benedek

Hungarian Academy of Sciences
benedek.marton@krtk.mta.hu

Outline

1. Introduction
   - Cooperative game theory
   - Literature
2. Sequential LP methods
   - Primal LP sequence
   - Dual LP sequence
3. Minimal tight set
4. Solving the large LPs
5. Geometry
6. Computational results

Introduction

Motivation

Beneficial to cooperate

- Cost allocation
  - Spanning tree games
- Combinatorial games
  - Flow games
  - Assignment and Matching games
- Political sciences
  - Weighted voting games (WVG)
  - Power indices
- Finance
  - Risk allocation games
- Network formation

Cooperative Game Theory

- Models various kinds of fair division or stable allocation problems
- Offers a plethora of solution concepts with attractive properties
- Most of which is unfortunately very hard to compute

Cooperative games

Definitions

- *Player set* $N = \{1, 2, \ldots, n\}$ with power set 2^N

Cooperative games

Definitions

- *Player set* $N = \{1, 2, \ldots, n\}$ with power set 2^N,
- Players form *coalitions* $S \subseteq N$

Cooperative games

Definitions

- *Player set* $N = \{1, 2, \ldots, n\}$ with power set 2^N,
- Players form *coalitions* $S \subseteq N$,
- *Value function* $v : 2^N \mapsto \mathbb{R}$

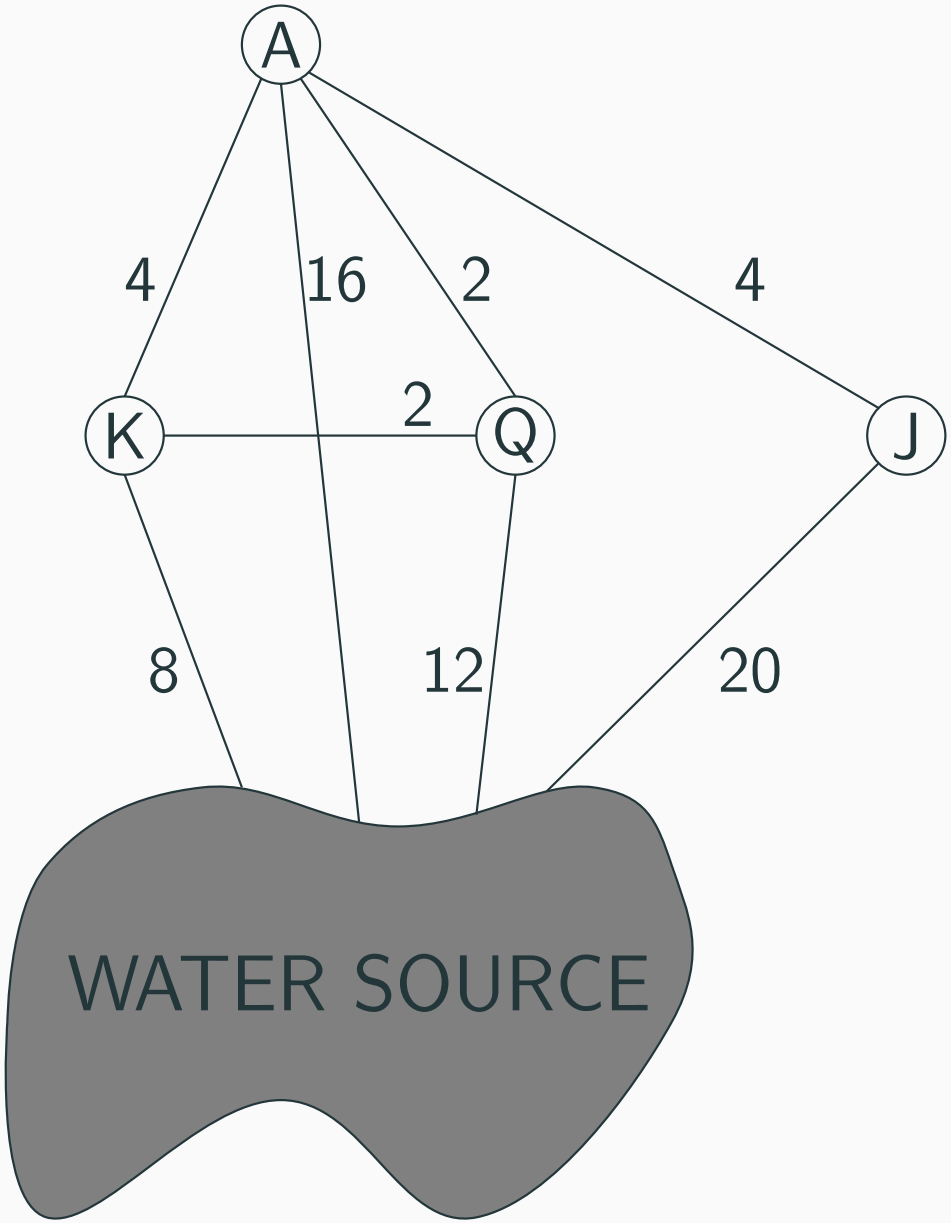
Cooperative games

Definitions

- *Player set* $N = \{1, 2, \ldots, n\}$ with power set 2^N,
- Players form *coalitions* $S \subseteq N$,
- *Value function* $v : 2^N \mapsto \mathbb{R}$
  - $v(\emptyset) = 0$
  - *grand coalition* N forms

$\Rightarrow$ **Aim:** divide $v(N)$ among the players *fairly*

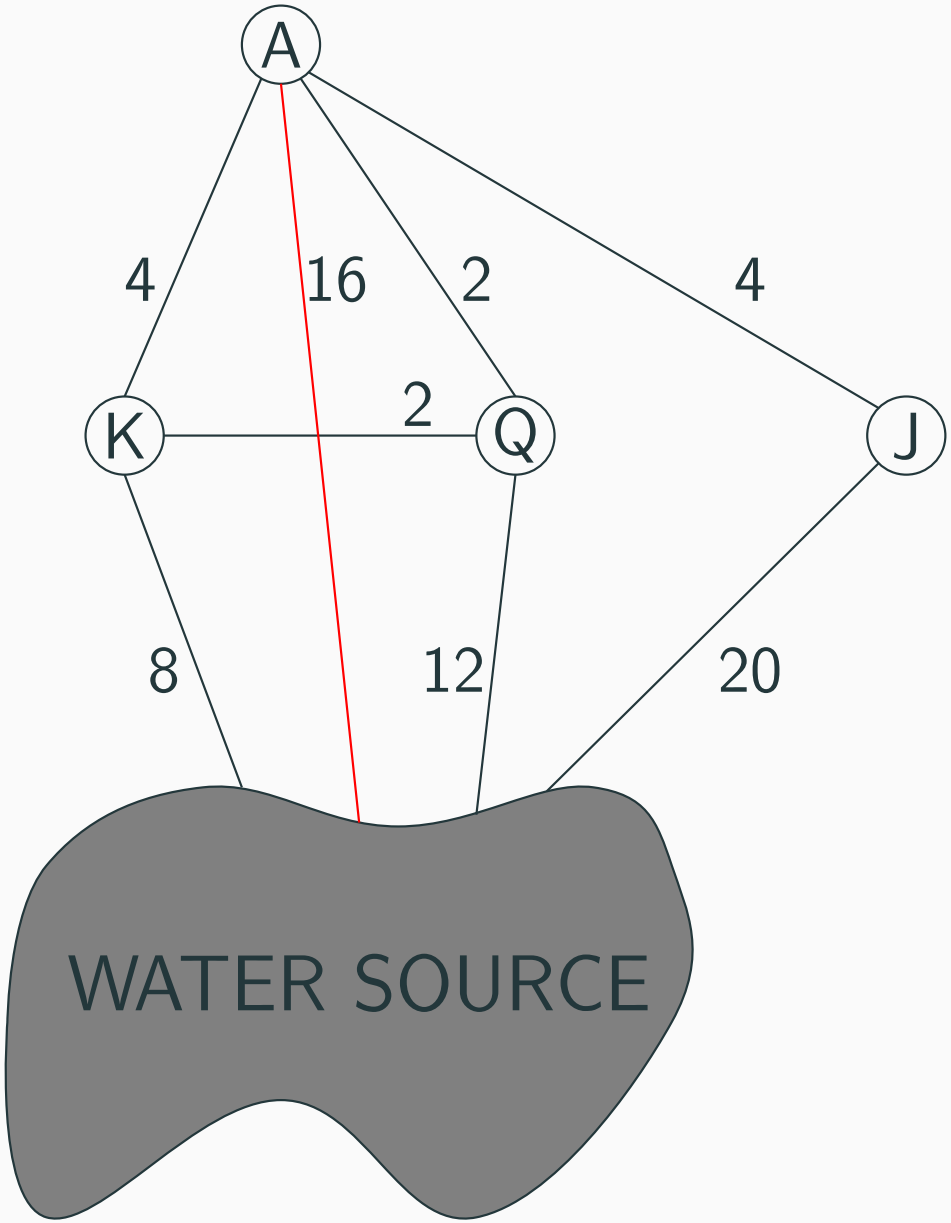
Example: cost allocation in a network



K:8 Q:12 A:16 J:20

- **Task #1:** agree to build a network that supplies every city

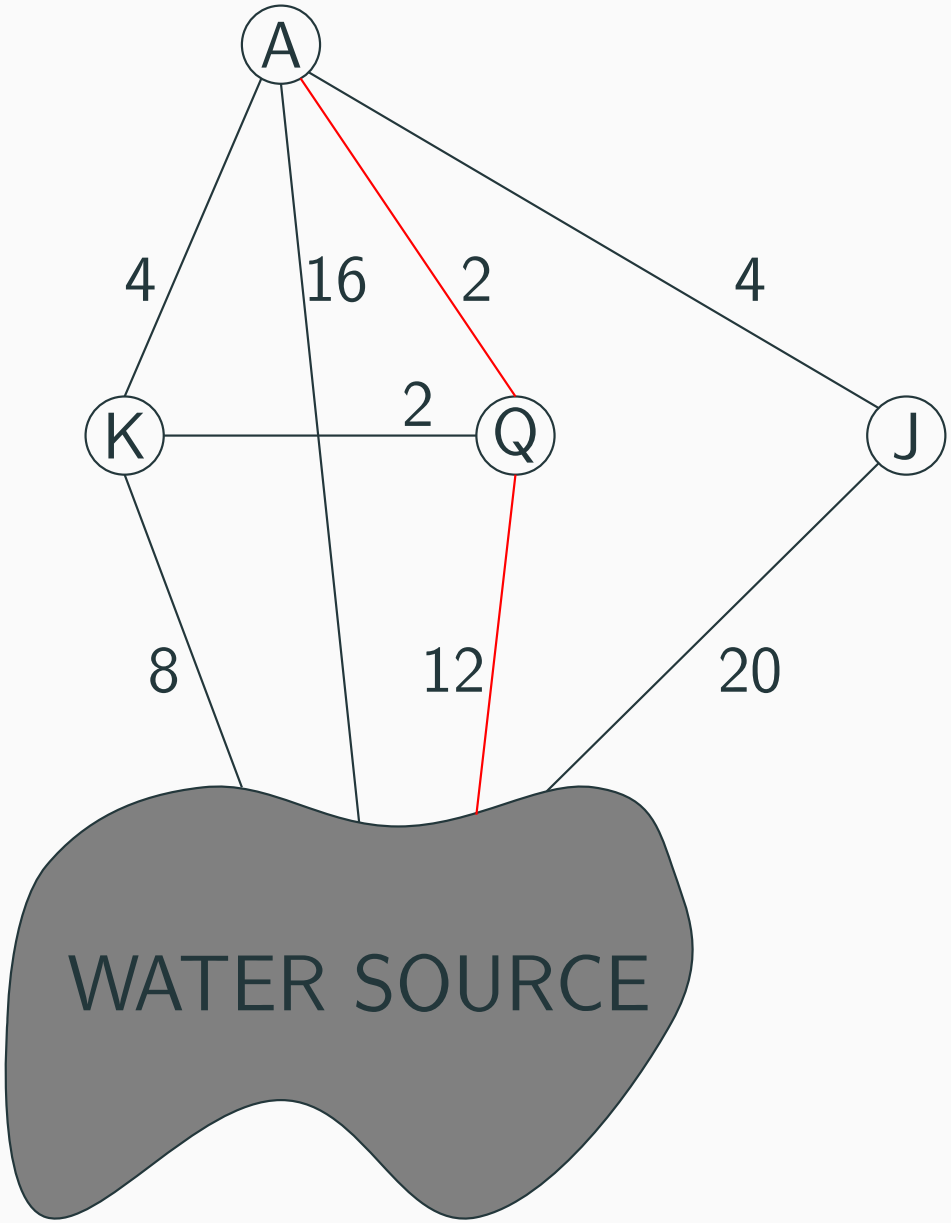
Example: cost allocation in a network



K:8 Q:12 A:16 J:20

- **Task #1:** agree to build a network that supplies every city
- Supply directly

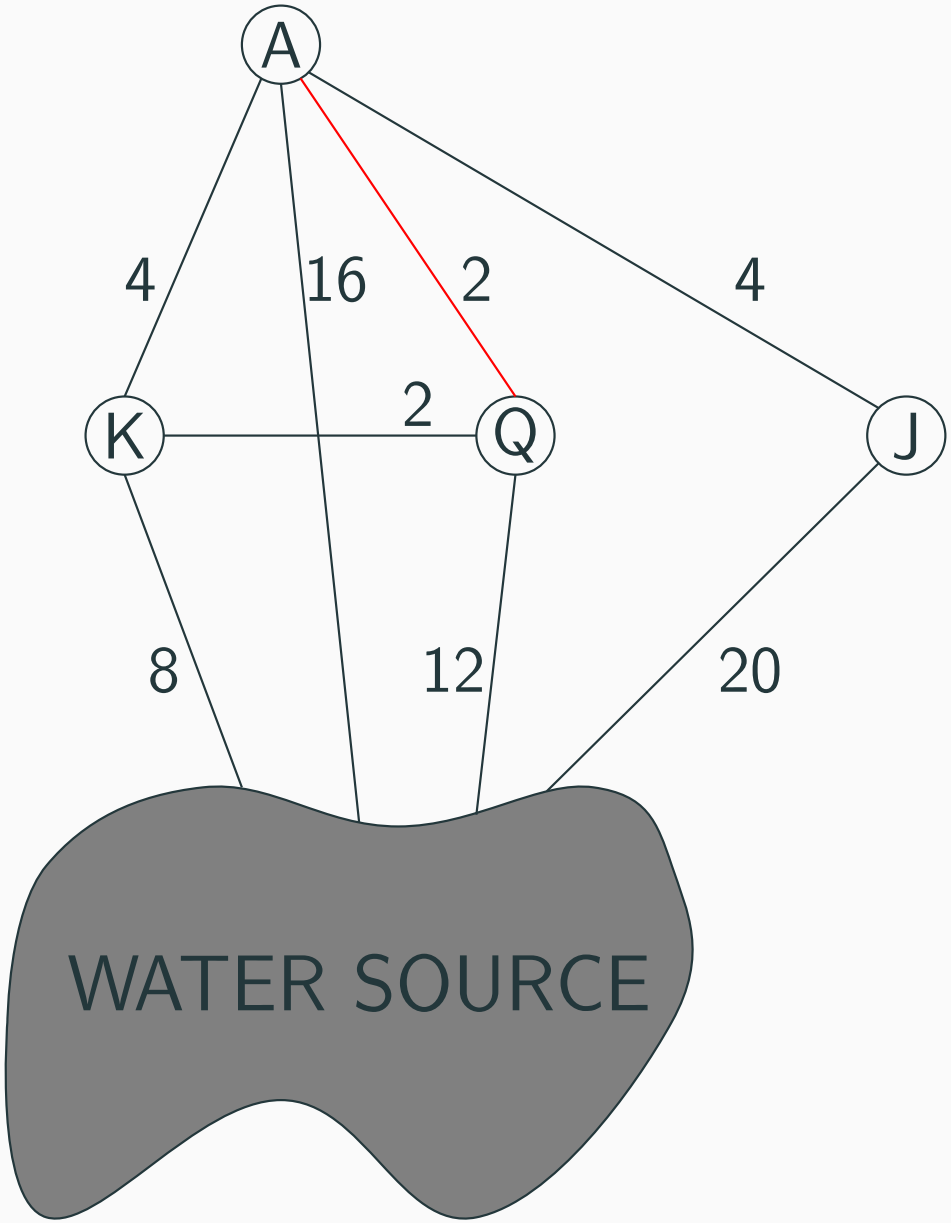
Example: cost allocation in a network



K:8 Q:12 A:16 J:20

- **Task #1:** agree to build a network that supplies every city
- Supply directly, or via a route of pipelines

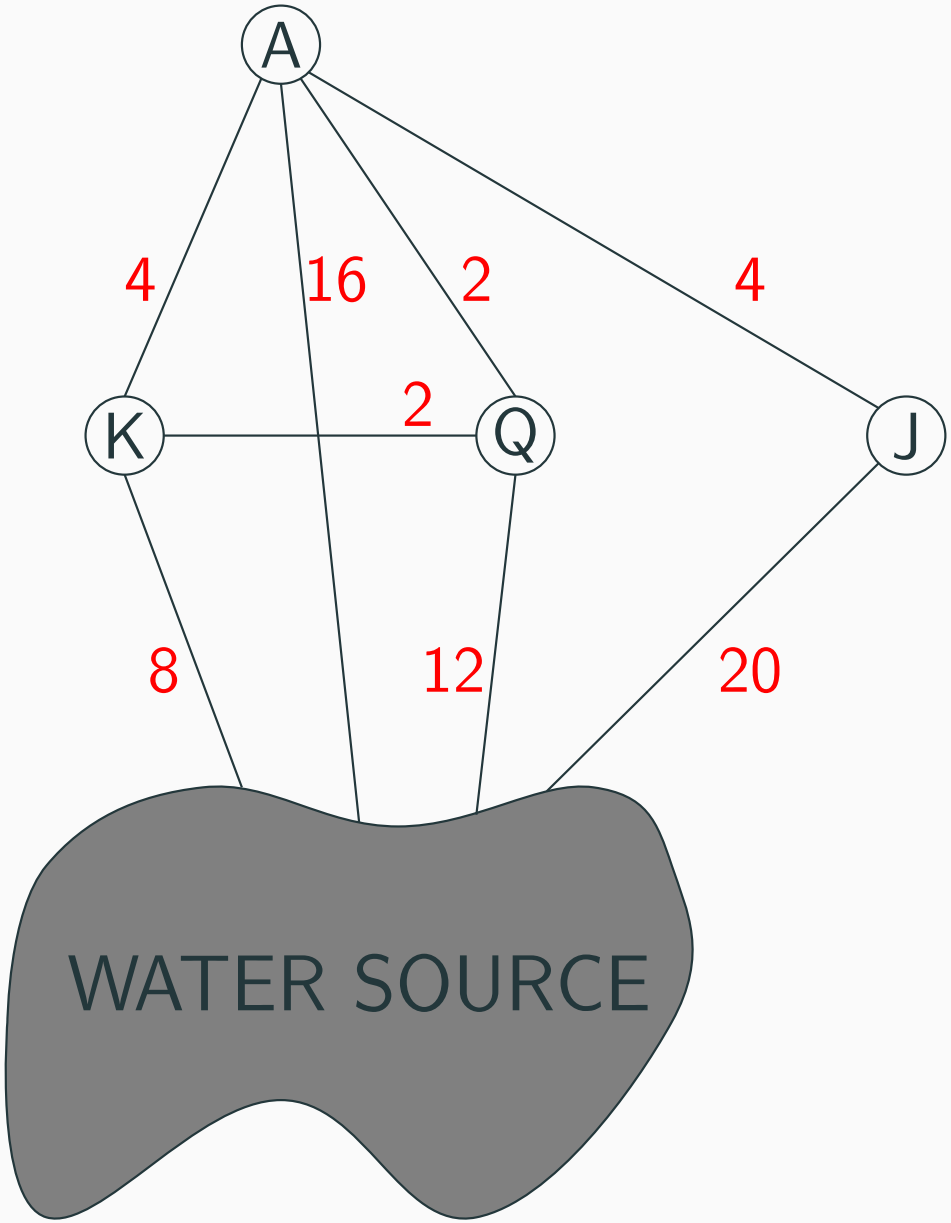
Example: cost allocation in a network



K:8 Q:12 A:16 J:20

- **Task #1:** agree to build a network that supplies every city
- Supply directly, or via a route of pipelines
- Pipeline building needs agreement

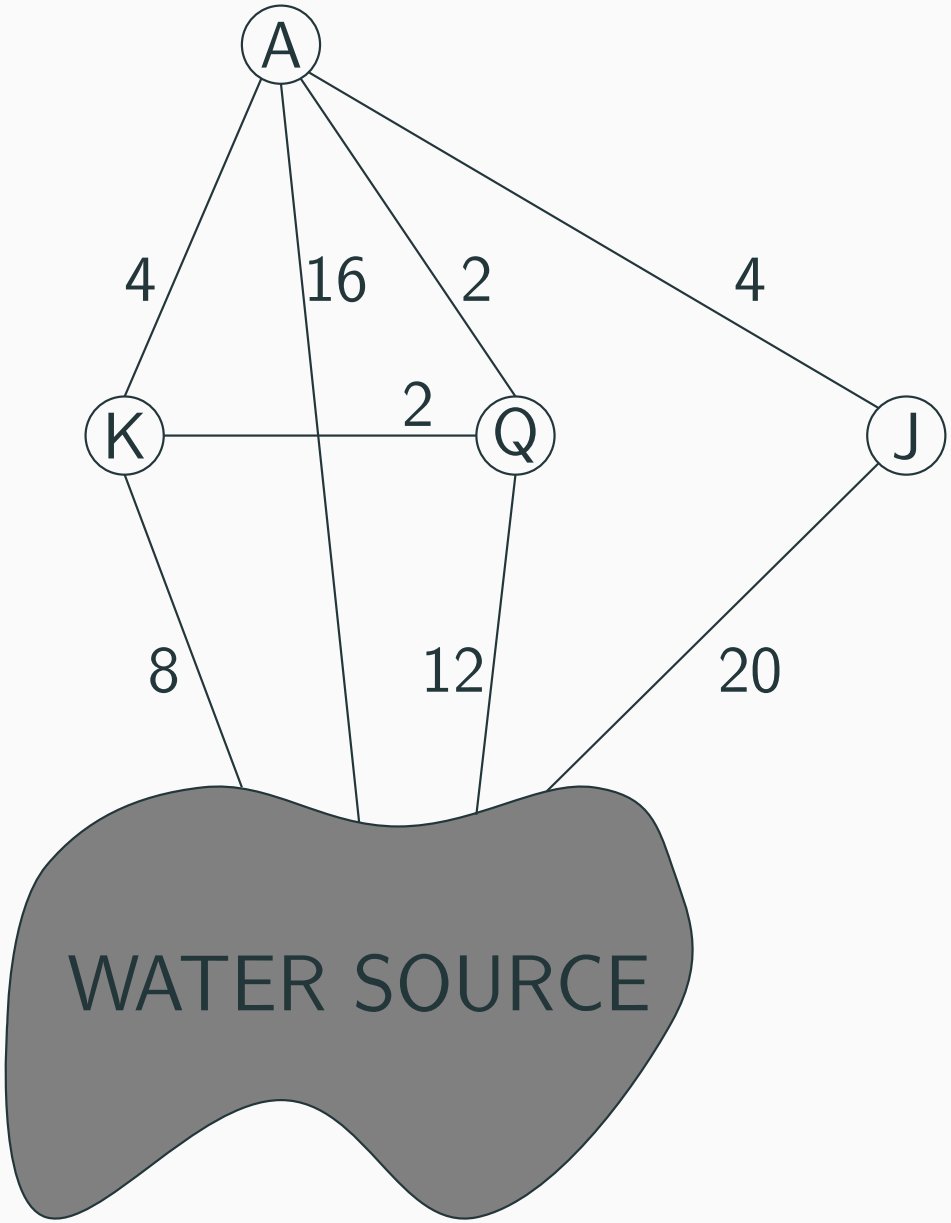
Example: cost allocation in a network



K:8 Q:12 A:16 J:20

- **Task #1:** agree to build a network that supplies every city
- Supply directly, or via a route of pipelines
- Pipeline building needs agreement
- Cost of each pipeline

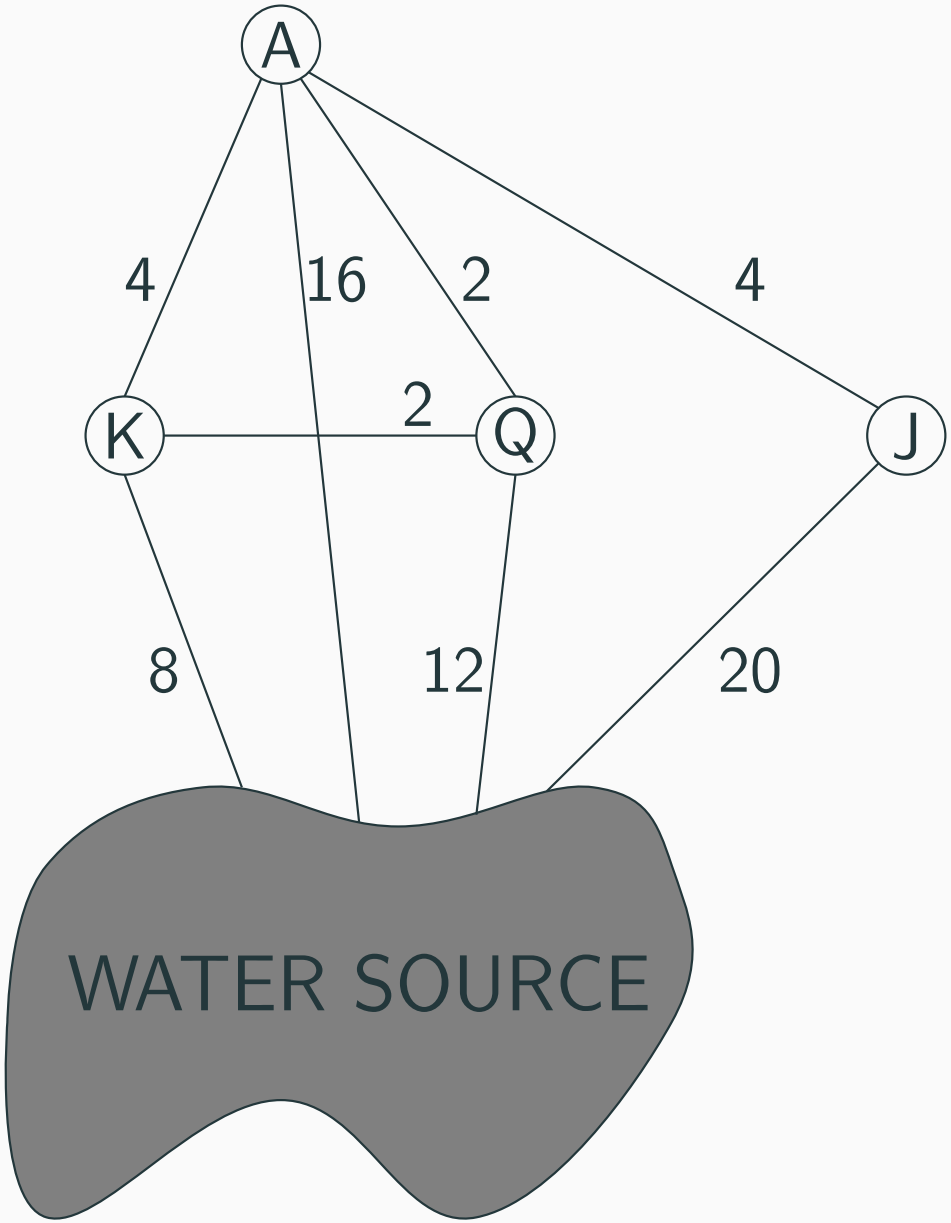
Example: cost allocation in a network



K:8 Q:12 A:16 J:20

- **Task #1:** agree to build a network that supplies every city
- Supply directly, or via a route of pipelines
- Pipeline building needs agreement
- Cost of each pipeline
- **Task #2:** agree on contributions

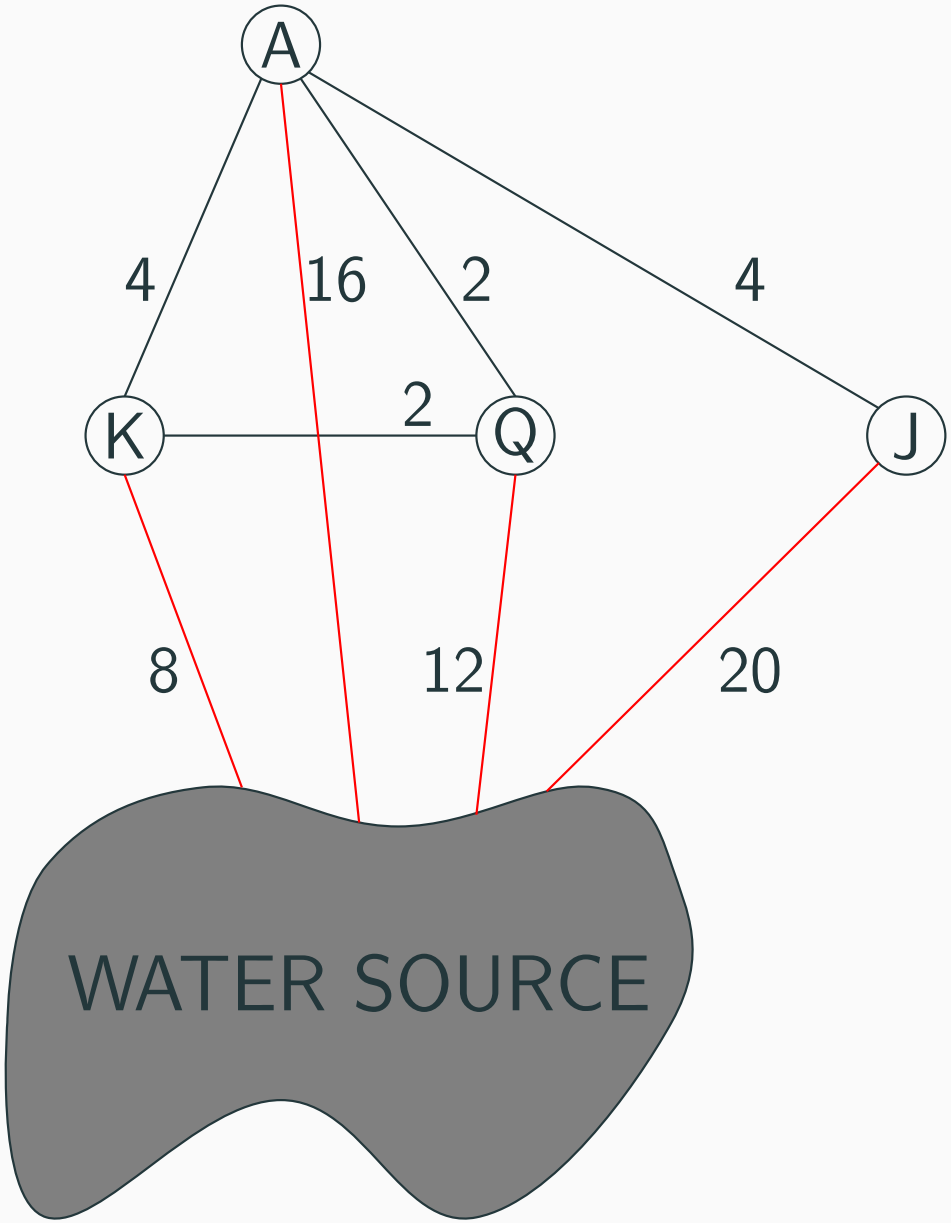
Example: cost allocation in a network



K:8 Q:12 A:16 J:20

- **Task #1:** agree to build a network that supplies every city
- Supply directly, or via a route of pipelines
- Pipeline building needs agreement
- Cost of each pipeline
- **Task #2:** agree on contributions; divide possible savings

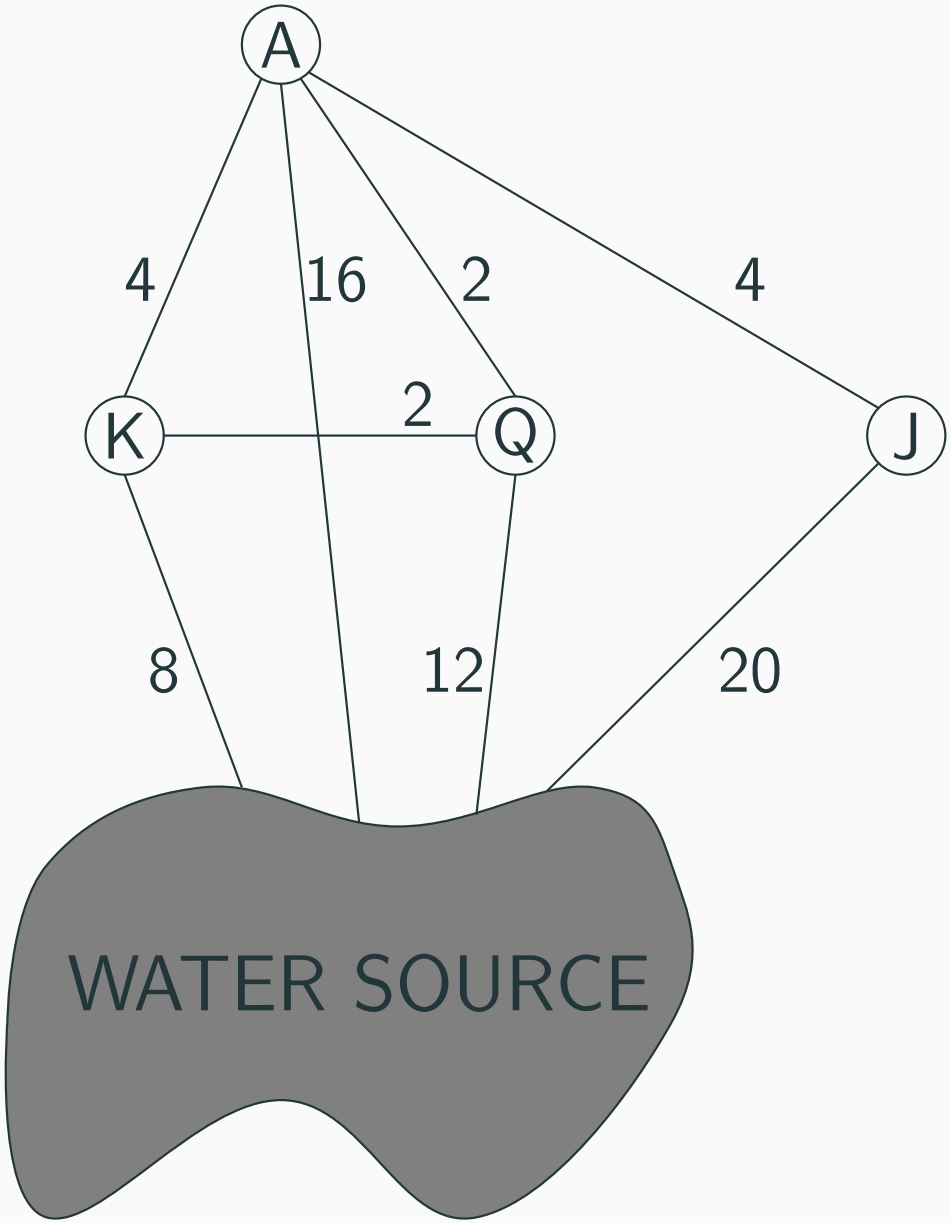
Example: cost allocation in a network



K:8 Q:12 A:16 J:20

- **Task #1:** agree to build a network that supplies every city
- Supply directly, or via a route of pipelines
- Pipeline building needs agreement
- Cost of each pipeline
- **Task #2:** agree on contributions; divide possible savings
- No agreement $\rightarrow$ network built using bank accounts

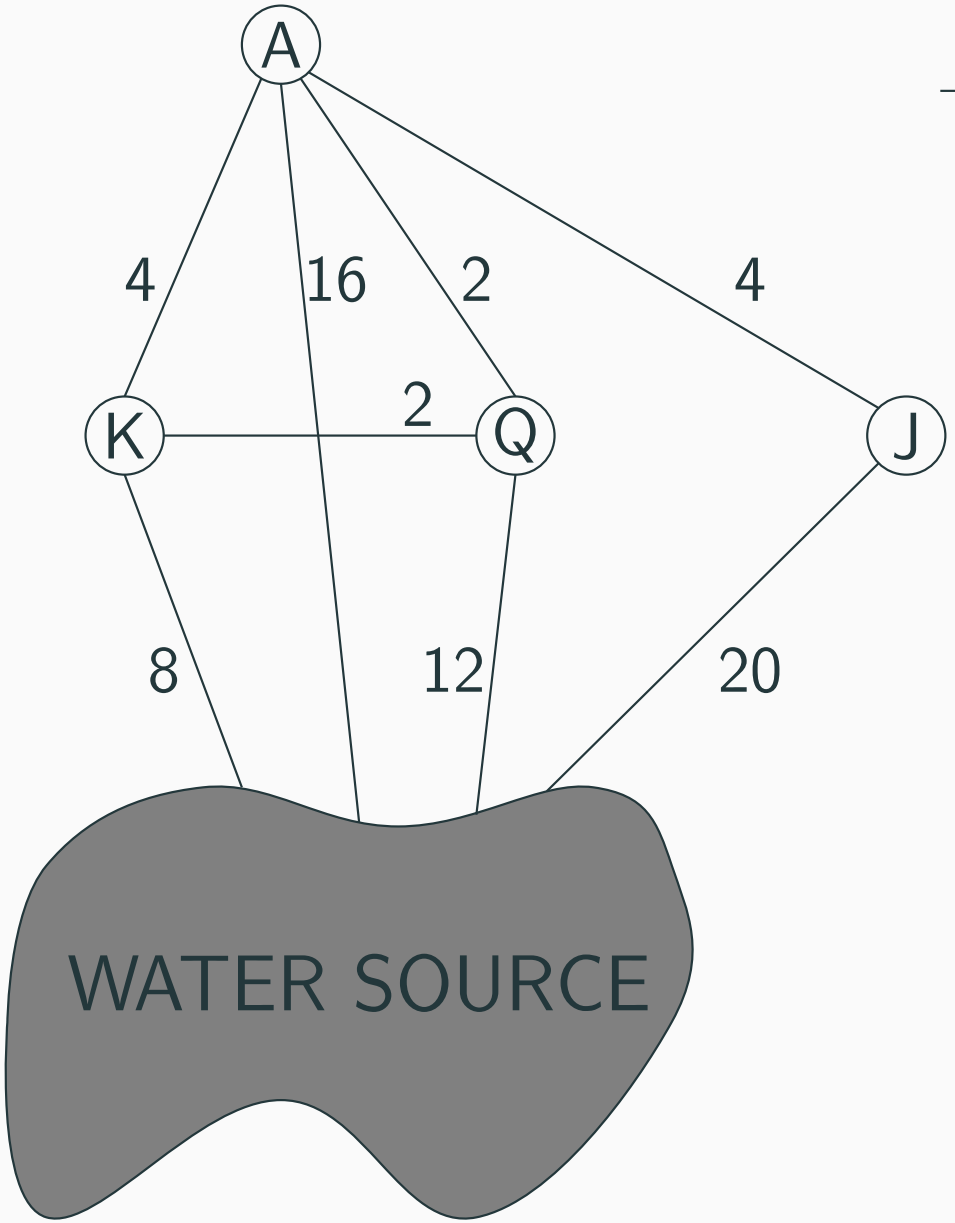
Example: cost allocation in a network



K:8 Q:12 A:16 J:20

- **Task #1:** agree to build a network that supplies every city
- Supply directly, or via a route of pipelines
- Pipeline building needs agreement
- Cost of each pipeline
- **Task #2:** agree on contributions; divide possible savings
- No agreement $\rightarrow$ network built using bank accounts

Example: cost allocation in a network



	cost	saving
K	8	0
Q	12	0
K,Q	10	10
A	16	0
A,K	12	12
A,Q	14	14
A,K,Q	12	24
J	20	0
K,J	28	0
Q,J	32	0
K,Q,J	30	10
A,J	20	16
A,K,J	16	28
A,Q,J	18	30
A,K,Q,J	16	40

Cooperative games

Definitions

- *Solution* $x \in \mathbb{R}^n$; notation $x(S) = \sum_{j \in S} x_j$

Cooperative games

Definitions

- *Solution* $x \in \mathbb{R}^n$; notation $x(S) = \sum_{j \in S} x_j$,
- *Allocation (preimputations)*: $x(N) = v(N)$; notation $pi(v)$

Cooperative games

Definitions

- *Solution* $x \in \mathbb{R}^n$; notation $x(S) = \sum_{j \in S} x_j$,
- *Allocation (preimputations)*: $x(N) = v(N)$; notation $pi(v)$
- *Individual rationality (imputations)*: $x_j \geq v(\{j\})\ \forall j \in N$; notation $ir(v)$

Cooperative games

Definitions

- *Solution* $x \in \mathbb{R}^n$; notation $x(S) = \sum_{j \in S} x_j$,
- *Allocation (preimputations)*: $x(N) = v(N)$; notation $pi(v)$
- *Individual rationality (imputations)*: $x_j \geq v(\{j\})\ \forall j \in N$; notation $ir(v)$
- **Excess** $E(S, x) = v(S) - x(S)$ for $x \in ir(v)$ and $\emptyset \neq S \subsetneq N$

Cooperative games

Definitions

- *Solution* $x \in \mathbb{R}^n$; notation $x(S) = \sum_{j \in S} x_j$,
- *Allocation (preimputations)*: $x(N) = v(N)$; notation $pi(v)$
- *Individual rationality (imputations)*: $x_j \geq v(\{j\})\ \forall j \in N$; notation $ir(v)$
- **Excess** $E(S, x) = v(S) - x(S)$ for $x \in ir(v)$ and $\emptyset \neq S \subsetneq N$ notation: $E(x) \in \mathbb{R}^{2^n - 2}$

Example: excesses as dissatisfactions

S	v
K	0
Q	0
K,Q	10
A	0
A,K	12
A,Q	14
A,K,Q	24
J	0
K,J	0
Q,J	0
K,Q,J	10
A,J	16
A,K,J	28
A,Q,J	30
A,K,Q,J	40

Example: excesses as dissatisfactions

S	v	ECS	excess
K	0	4	-4
Q	0	8	-8
K,Q	10	12	-2
A	0	12	-12
A,K	12	16	-4
A,Q	14	20	-6
A,K,Q	24	24	0
J	0	16	-16
K,J	0	20	-20
Q,J	0	24	-24
K,Q,J	10	28	-18
A,J	16	28	-12
A,K,J	28	32	-4
A,Q,J	30	36	-6
A,K,Q,J	40	40	

In (K, Q, A, J) order

Equal cost sharing (ECS)

- $(4, 4, 4, 4)$ cost
- $(4, 8, 12, 16)$ saving

Example: excesses as dissatisfactions

S	v	ECS	excess	ESD	excess
K	0	4	-4	10	-10
Q	0	8	-8	10	-10
K,Q	10	12	-2	20	-10
A	0	12	-12	10	-10
A,K	12	16	-4	20	-8
A,Q	14	20	-6	20	-6
A,K,Q	24	24	0	30	-6
J	0	16	-16	10	-10
K,J	0	20	-20	20	-20
Q,J	0	24	-24	20	-20
K,Q,J	10	28	-18	30	-20
A,J	16	28	-12	20	-4
A,K,J	28	32	-4	30	-2
A,Q,J	30	36	-6	30	0
A,K,Q,J	40	40		40	

In (K, Q, A, J) order

Equal cost sharing (ECS)

- $(4, 4, 4, 4)$ cost
- $(4, 8, 12, 16)$ saving

Equal surplus division (ESD)

- $(-2, 2, 6, 10)$ cost
- $(10, 10, 10, 10)$ saving

Solution concepts

- Coalitionally rational outcomes form the core:
  Core [Shapley, 1955] $c(v) = \{x \in ir(v) : E(S, x) \leq 0 \ \forall S \subsetneq N\}$,

Solution concepts

- Coalitionally rational outcomes form the core:
  Core [Shapley, 1955] $c(v) = \{x \in ir(v) : E(S, x) \leq 0 \ \forall S \subsetneq N\}$,
  could be empty, or we could have $|c(v)| > 1$

Solution concepts

- Coalitionally rational outcomes form the core:
  Core [Shapley, 1955] $c(v) = \{x \in ir(v) : E(S, x) \leq 0 \; \forall S \subsetneq N\}$,
  could be empty, or we could have $|c(v)| > 1$
- Shapley-value [Shapley, 1953]:

$$\phi(v)_i = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(n - |S| - 1)!}{n!} \left(v(S \cup \{i\}) - v(S)\right), \quad \forall i \in N$$

Cooperative games

Solution concepts

- Coalitionally rational outcomes form the core:
  Core [Shapley, 1955] $c(v) = \{x \in ir(v) : E(S, x) \leq 0 \ \forall S \subsetneq N\}$,
  could be empty, or we could have $|c(v)| > 1$
- Shapley-value [Shapley, 1953]:

$$\phi(v)_i = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(n - |S| - 1)!}{n!} \left(v(S \cup \{i\}) - v(S)\right), \quad \forall i \in N$$

- Relaxation/restriction of the core:
  ε-*Core* $c_\varepsilon(v) = \{x \in ir(v) : E(S, x) \leq \varepsilon \ \forall S \subsetneq N\}$,

Cooperative games

Solution concepts

- Coalitionally rational outcomes form the core:
  Core [Shapley, 1955] $c(v) = \{x \in ir(v) : E(S, x) \leq 0 \ \forall S \subsetneq N\}$,
  could be empty, or we could have $|c(v)| > 1$
- Shapley-value [Shapley, 1953]:

$$\phi(v)_i = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(n - |S| - 1)!}{n!} \left(v(S \cup \{i\}) - v(S)\right), \quad \forall i \in N$$

- Relaxation/restriction of the core:
  ε-*Core* $c_\varepsilon(v) = \{x \in ir(v) : E(S, x) \leq \varepsilon \ \forall S \subsetneq N\}$,
  for large enough ε, $c_\varepsilon(v) \neq \emptyset$

Cooperative games

Solution concepts

- Coalitionally rational outcomes form the core:
  Core [Shapley, 1955] $c(v) = \{x \in ir(v) : E(S,x) \leq 0 \ \forall S \subsetneq N\}$,
  could be empty, or we could have $|c(v)| > 1$
- Shapley-value [Shapley, 1953]:

$$\phi(v)_i = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(n-|S|-1)!}{n!} \left(v(S \cup \{i\}) - v(S)\right), \quad \forall i \in N$$

- Relaxation/restriction of the core:
  ε-*Core* $c_\varepsilon(v) = \{x \in ir(v) : E(S,x) \leq \varepsilon \ \forall S \subsetneq N\}$,
  for large enough ε, $c_\varepsilon(v) \neq \emptyset$
- Interesting: smallest $\varepsilon^*(v)$ such that $c^*_\varepsilon(v) \neq \emptyset$
  Least core $lc(v) = c_{\varepsilon^*}(v)$, where $\varepsilon^* = \min\{\varepsilon : c_\varepsilon(v) \neq \emptyset\}$,
  non-emptiness guaranteed, but uniqueness still not

Nucleolus

Taking the idea of the least core to more levels $\Rightarrow$ nucleolus

1. Minimise the largest dissatisfaction $\Rightarrow$ least core

Nucleolus

Taking the idea of the least core to more levels $\Rightarrow$ nucleolus

1. Minimise the largest dissatisfaction $\Rightarrow$ least core
2. Minimise the number of coalitions receiving that $\Rightarrow$ interior of the least core

Nucleolus

Taking the idea of the least core to more levels $\Rightarrow$ nucleolus

1. Minimise the largest dissatisfaction $\Rightarrow$ least core
2. Minimise the number of coalitions receiving that $\Rightarrow$ interior of the least core
3. Minimise the second largest dissatisfaction (among remaining)

Nucleolus

Taking the idea of the least core to more levels $\Rightarrow$ nucleolus

1. Minimise the largest dissatisfaction $\Rightarrow$ least core
2. Minimise the number of coalitions receiving that $\Rightarrow$ interior of the least core
3. Minimise the second largest dissatisfaction (among remaining)
4. Minimise the number of coalitions receiving that

Nucleolus

Taking the idea of the least core to more levels $\Rightarrow$ nucleolus

1. Minimise the largest dissatisfaction $\Rightarrow$ least core
2. Minimise the number of coalitions receiving that $\Rightarrow$ interior of the least core
3. Minimise the second largest dissatisfaction (among remaining)
4. Minimise the number of coalitions receiving that

   . . . and so on so forth . . .

Nucleolus

Taking the idea of the least core to more levels $\Rightarrow$ nucleolus

1. Minimise the largest dissatisfaction $\Rightarrow$ least core
2. Minimise the number of coalitions receiving that $\Rightarrow$ interior of the least core
3. Minimise the second largest dissatisfaction (among remaining)
4. Minimise the number of coalitions receiving that

   . . . and so on so forth . . .

   Until: uniqueness

Nucleolus

Nucleolus [Schmeidler, 1969]

- Vector of excesses
  $E(x) = [E(S_1, x), E(S_2, x), \ldots, E(S_{2^n-2}, x)] \in \mathbb{R}^{2^n-2}$,

Nucleolus

Nucleolus [Schmeidler, 1969]

- Vector of excesses
  $E(x) = [E(S_1, x), E(S_2, x), \ldots, E(S_{2^n-2}, x)] \in \mathbb{R}^{2^n-2}$,
- Non-increasingly ordered by $\Theta : \mathbb{R}^m \mapsto \mathbb{R}^m$
  $\Theta(y)_1 \geq \Theta(y)_2 \geq \cdots \geq \Theta(y)_m$ for any $y \in \mathbb{R}^m$

Nucleolus

Nucleolus [Schmeidler, 1969]

- Vector of excesses
  $E(x) = [E(S_1, x), E(S_2, x), \ldots, E(S_{2^n-2}, x)] \in \mathbb{R}^{2^n-2}$,
- Non-increasingly ordered by $\Theta : \mathbb{R}^m \mapsto \mathbb{R}^m$
  $\Theta(y)_1 \geq \Theta(y)_2 \geq \cdots \geq \Theta(y)_m$ for any $y \in \mathbb{R}^m$
- The nucleolus is

$$\nu \in ir(v): \quad \Theta(E(\nu)) \leq_L \Theta(E(x)) \quad \forall x \in ir(v),$$

  where
  - $x, y \in \mathbb{R}^m : x \leq_L y \Leftrightarrow$ either $x = y$ or $\exists j \in \{0, 1, \ldots, m-1\}$:
    $x_{j+1} < y_{j+1}$, and if $j > 0$ then $x_k = y_k \ \forall k \in \{1, \ldots, j\}$

Example: excesses as dissatisfactions

S	v	avg	excess
K	0	7	-7
Q	0	9	-9
K,Q	10	16	-6
A	0	11	-11
A,K	12	18	-6
A,Q	14	20	-6
A,K,Q	24	27	-3
J	0	13	-13
K,J	0	20	-20
Q,J	0	22	-22
K,Q,J	10	29	-19
A,J	16	24	-8
A,K,J	28	31	-3
A,Q,J	30	33	-3
A,K,Q,J	40	40	

In (K, Q, A, J) order

Average of ECS and ESD (avg)

- $(1, 3, 5, 7)$ cost
- $(7, 9, 11, 13)$ saving

Example: excesses as dissatisfactions

S	v	avg	excess	ν	excess
K	0	7	-7	6	-6
Q	0	9	-9	8	-8
K,Q	10	16	-6	14	-4
A	0	11	-11	18	-18
A,K	12	18	-6	24	-12
A,Q	14	20	-6	26	-12
A,K,Q	24	27	-3	32	-8
J	0	13	-13	8	-8
K,J	0	20	-20	14	-14
Q,J	0	22	-22	16	-16
K,Q,J	10	29	-19	22	-12
A,J	16	24	-8	26	-10
A,K,J	28	31	-3	32	-4
A,Q,J	30	33	-3	34	-4
A,K,Q,J	40	40		40	

In (K, Q, A, J) order

Average of ECS and ESD (avg)

- $(1, 3, 5, 7)$ cost
- $(7, 9, 11, 13)$ saving

Nucleolus (ν)

"Feed the hungriest first."

- $(2, 4, -2, 12)$ cost
- $(6, 8, 18, 8)$ saving

Example: excesses as dissatisfactions

S	v	avg	excess	ν	excess
K	0	7	-7	6	-6
Q	0	9	-9	8	-8
K,Q	10	16	-6	14	-4
A	0	11	-11	18	-18
A,K	12	18	-6	24	-12
A,Q	14	20	-6	26	-12
A,K,Q	24	27	-3	32	-8
J	0	13	-13	8	-8
K,J	0	20	-20	14	-14
Q,J	0	22	-22	16	-16
K,Q,J	10	29	-19	22	-12
A,J	16	24	-8	26	-10
A,K,J	28	31	-3	32	-4
A,Q,J	30	33	-3	34	-4
A,K,Q,J	40	40		40	

In (K, Q, A, J) order

Average of ECS and ESD (avg)

- $(1, 3, 5, 7)$ cost
- $(7, 9, 11, 13)$ saving

Nucleolus (ν)

"Feed the hungriest first."

- $(2, 4, -2, 12)$ cost
- $(6, 8, 18, 8)$ saving

Surprising application

(pre)nucleolus

Nucleolus vs prenucleolus: $\nu \in pi(v)$ instead of $\nu \in ir(v)$

(pre)nucleolus

Nucleolus vs prenucleolus: $\nu \in pi(v)$ instead of $\nu \in ir(v)$

Desirable properties

For any cooperative game (N, v) the prenucleolus ν:

- ν exists
- ν is unique
- $\nu \in c_\varepsilon(v)$ for all $\varepsilon \geq \varepsilon^*$
- consequently $\nu \in lc(v)$
- ν is a continuous function of v

(pre)nucleolus

Nucleolus vs prenucleolus: $\nu \in pi(v)$ instead of $\nu \in ir(v)$

Desirable properties

For any cooperative game (N, v) the prenucleolus ν:

- ν exists
- ν is unique
- $\nu \in c_\varepsilon(v)$ for all $\varepsilon \geq \varepsilon^*$
- consequently $\nu \in lc(v)$
- ν is a continuous function of v

However...

$$\begin{aligned} \text{lex} \min_x \quad & \Theta(E(x)) \\ \text{s.t.} \quad & x(N) = v(N) \end{aligned}$$

Span of coalitions

Definition

Characteristic vector $e(S) \in \{0,1\}^n : e(S)_j = 1 \iff j \in S$

Coalition $S \in \text{span}(\mathcal{B})$ if $\exists \beta \in \mathbb{R}^{|\mathcal{B}|}$:

$$e(S) = \sum_{Q \in \mathcal{B}} \beta_Q e(Q)$$

Rank of collection of coalitions can be defined straightforwardly:

$\text{rank}(\mathcal{B}) = r$ if $\exists \mathcal{R} \subseteq \mathcal{B}$ *smallest* such that $|\mathcal{R}| = r$ and
$S \in \text{span}(\mathcal{B}) \iff S \in \text{span}(\mathcal{R})$

Literature

Verification:

- Kohlberg criteria [Kohlberg, 1971]
- Nonlinear approximation [Kido, 2008]

Literature

Verification:

- Kohlberg criteria [Kohlberg, 1971]
- Nonlinear approximation [Kido, 2008]

Table 1: Single LP methods

	type	LPs	rows	columns
[Kohlberg, 1972]	primal	1	$(2^n)!$	n
[Owen, 1974]	primal	1	$\mathcal{O}(4^n)$	$\mathcal{O}(2^n)$
[Puerto and Perea, 2013]	primal	1	$\mathcal{O}(4^n)$	$\mathcal{O}(4^n)$

Literature

Table 2: Sequential LP methods

	type	LPs	rows	columns
[Maschler et al., 1979]	primal	$\mathcal{O}(4^n)$	$\mathcal{O}(2^n)$	$n+1$
[Dragan, 1981]	dual	$n-1$	$\mathcal{O}(n)$	$\mathcal{O}(2^n)$
[Sankaran, 1991]	primal	$\mathcal{O}(2^n)$	$\mathcal{O}(2^n)$	$n+1$
[Solymosi, 1993]	primal dual	$n-1$	$\mathcal{O}(2^n)$	$n+1$
	dual	$n-1$	$n+1$	$\mathcal{O}(2^n)$
[Potters et al., 1996]	(primal-dual)	$n-1$	$\mathcal{O}(2^n)$	$\mathcal{O}(2^n)$
[Derks and Kuipers, 1997]	dual	$n-1$	$n+1$	$\mathcal{O}(2^n)$
[Nguyen and Thomas, 2016]	primal primal	$n-1$	$\mathcal{O}(2^n)$	$n+1$
[Benedek et al., 2020]	primal dual	$n-1$	$\mathcal{O}(2^n)$	$n+1$

Sequential LP methods

Primal LP sequence

$$
\begin{aligned}
\min_{x^{(1)},\varepsilon^{(1)}} \quad & \varepsilon^{(1)} & & & & \\
\text{s.t.} \quad & \varepsilon^{(1)} + x^{(1)}(S) & \geq v(S) & \quad \forall S \subsetneq N, S \neq \emptyset & & (\mathrm{P}^{(1)}) \\
& x^{(1)}(N) & = v(N) & & &
\end{aligned}
$$

Primal LP sequence

$$
\begin{aligned}
\min_{x^{(1)},\varepsilon^{(1)}} \quad & \varepsilon^{(1)} & & & \\
\text{s.t.} \quad & \varepsilon^{(1)} + x^{(1)}(S) & \geq v(S) & \quad \forall S \subsetneq N, S \neq \emptyset & \qquad (\mathrm{P}^{(1)}) \\
& x^{(1)}(N) & = v(N) &
\end{aligned}
$$

Problem #1: how to solve this?

Primal LP sequence

$$
\begin{aligned}
\min_{x^{(1)},\varepsilon^{(1)}} \quad & \varepsilon^{(1)} & & & \\
\text{s.t.} \quad & \varepsilon^{(1)} + x^{(1)}(S) & \geq v(S) & \quad \forall S \subsetneq N, S \neq \emptyset & \qquad (\mathrm{P}^{(1)}) \\
& x^{(1)}(N) & = v(N) &
\end{aligned}
$$

Problem #1: how to solve this? Moreover, how to find not *any*, but particular solutions satisfying certain properties?

Primal LP sequence

$$\begin{aligned} \min_{x^{(1)},\varepsilon^{(1)}} \quad & \varepsilon^{(1)} & & & \\ \text{s.t.} \quad & \varepsilon^{(1)} + x^{(1)}(S) & \geq v(S) & \quad \forall S \subsetneq N, S \neq \emptyset & (\mathrm{P}^{(1)}) \\ & x^{(1)}(N) & = v(N) & & \end{aligned}$$

Problem #1: how to solve this? Moreover, how to find not *any*, but particular solutions satisfying certain properties?

Definition (tight set)

Given primal feasible $(x^{(1)}, \epsilon^{(1)}) \in \mathcal{P}^{(1)}$ we define $\mathcal{T}^{(1)}(x^{(1)}, \epsilon^{(1)})$ to be the set of tight constraints:

$$\epsilon^{(1)} + x^{(1)}(S) = v(S) \quad \forall S \in \mathcal{T}^{(1)}(x^{(1)}, \epsilon^{(1)})$$

Notation: $\mathcal{T}^{(1)}(x^{(1)})$ for optimal $x^{(1)} \in \mathcal{P}^{(1)*}$

Primal LP sequence

LP sequence strategy: iteratively finding *certain* tight coalitions and setting the corresponding inequality constraints as equality constraints

Primal LP sequence

LP sequence strategy: iteratively finding *certain* tight coalitions and setting the corresponding inequality constraints as equality constraints

Importance of tight set sizes

- Suppose $x, y \in \mathcal{P}^{(1)*}$ with $j = |\mathcal{T}^{(1)}(x)| < |\mathcal{T}^{(1)}(y)|$.

Primal LP sequence

LP sequence strategy: iteratively finding *certain* tight coalitions and setting the corresponding inequality constraints as equality constraints

Importance of tight set sizes

- Suppose $x, y \in \mathcal{P}^{(1)*}$ with $j = |\mathcal{T}^{(1)}(x)| < |\mathcal{T}^{(1)}(y)|$.
- In the first j elements: $\Theta(E(x))_i = \Theta(E(y))_i = \varepsilon^{(1)*}$ for $i \leq j$

Primal LP sequence

LP sequence strategy: iteratively finding *certain* tight coalitions and setting the corresponding inequality constraints as equality constraints

Importance of tight set sizes

- Suppose $x, y \in \mathcal{P}^{(1)*}$ with $j = |\mathcal{T}^{(1)}(x)| < |\mathcal{T}^{(1)}(y)|$.
- In the first j elements: $\Theta(E(x))_i = \Theta(E(y))_i = \varepsilon^{(1)*}$ for $i \leq j$ while $\Theta(E(x))_{j+1} < \varepsilon^{(1)*} = \Theta(E(y))_{j+1}$.

Primal LP sequence

LP sequence strategy: iteratively finding *certain* tight coalitions and setting the corresponding inequality constraints as equality constraints

Importance of tight set sizes

- Suppose $x, y \in \mathcal{P}^{(1)*}$ with $j = |\mathcal{T}^{(1)}(x)| < |\mathcal{T}^{(1)}(y)|$.
- In the first j elements: $\Theta(E(x))_i = \Theta(E(y))_i = \varepsilon^{(1)*}$ for $i \leq j$ while $\Theta(E(x))_{j+1} < \varepsilon^{(1)*} = \Theta(E(y))_{j+1}$.
- It follows from $\Theta(E(y)) \not\leq_L \Theta(E(x))$ that $y \neq \nu$.

Primal LP sequence

LP sequence strategy: iteratively finding *certain* tight coalitions and setting the corresponding inequality constraints as equality constraints

Importance of tight set sizes

- Suppose $x, y \in \mathcal{P}^{(1)*}$ with $j = |\mathcal{T}^{(1)}(x)| < |\mathcal{T}^{(1)}(y)|$.
- In the first j elements: $\Theta(E(x))_i = \Theta(E(y))_i = \varepsilon^{(1)*}$ for $i \leq j$ while $\Theta(E(x))_{j+1} < \varepsilon^{(1)*} = \Theta(E(y))_{j+1}$.
- It follows from $\Theta(E(y)) \not\leq_L \Theta(E(x))$ that $y \neq \nu$.
- $|\mathcal{T}^{(1)}(\nu)| \leq |\mathcal{T}^{(1)}(x)|$ for any $x \in \mathcal{P}^{(1)*}$.

Primal LP sequence

LP sequence strategy: iteratively finding *certain* tight coalitions and setting the corresponding inequality constraints as equality constraints

Importance of tight set sizes

- Suppose $x, y \in \mathcal{P}^{(1)*}$ with $j = |\mathcal{T}^{(1)}(x)| < |\mathcal{T}^{(1)}(y)|$.
- In the first j elements: $\Theta(E(x))_i = \Theta(E(y))_i = \varepsilon^{(1)*}$ for $i \leq j$ while $\Theta(E(x))_{j+1} < \varepsilon^{(1)*} = \Theta(E(y))_{j+1}$.
- It follows from $\Theta(E(y)) \not\leq_L \Theta(E(x))$ that $y \neq \nu$.
- $|\mathcal{T}^{(1)}(\nu)| \leq |\mathcal{T}^{(1)}(x)|$ for any $x \in \mathcal{P}^{(1)*}$.

Lemma 1 (minimal tight set)

There exists a unique tight set $\mathcal{T}^{(1)*}$, such that $|\mathcal{T}^{(1)*}| \leq |\mathcal{T}^{(1)}(x)|$ for all $x \in \mathcal{P}^{(1)*}$. Moreover $\mathcal{T}^{(1)*} \subseteq \mathcal{T}^{(1)}(x)$.

$\mathcal{T}^{(1)*} = \mathcal{T}^{(1)}(\nu)$ is called the *minimal tight set*.

Primal LP sequence

LP sequence strategy: iteratively finding *certain* tight coalitions and setting the corresponding inequality constraints as equality constraints

Importance of tight set sizes

- Suppose $x, y \in \mathcal{P}^{(1)*}$ with $j = |\mathcal{T}^{(1)}(x)| < |\mathcal{T}^{(1)}(y)|$.
- In the first j elements: $\Theta(E(x))_i = \Theta(E(y))_i = \varepsilon^{(1)*}$ for $i \leq j$ while $\Theta(E(x))_{j+1} < \varepsilon^{(1)*} = \Theta(E(y))_{j+1}$.
- It follows from $\Theta(E(y)) \not\leq_L \Theta(E(x))$ that $y \neq \nu$.
- $|\mathcal{T}^{(1)}(\nu)| \leq |\mathcal{T}^{(1)}(x)|$ for any $x \in \mathcal{P}^{(1)*}$.

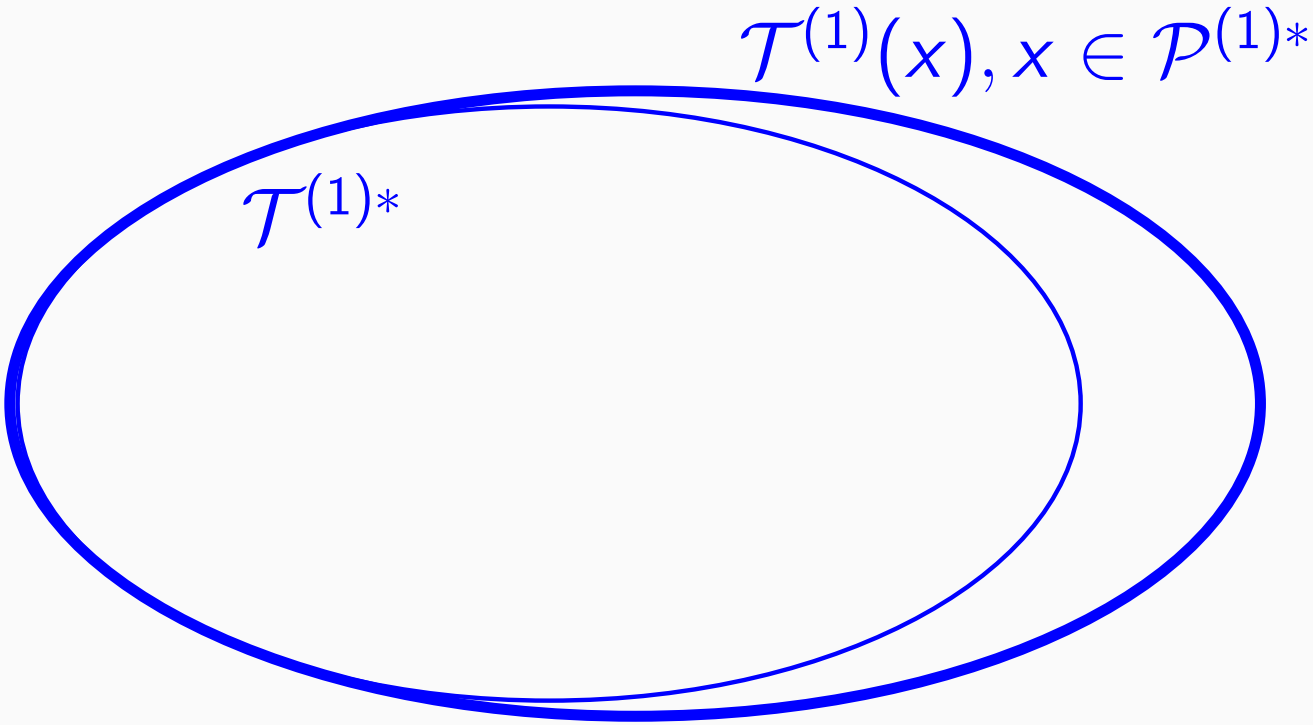
Lemma 1 (minimal tight set)

There exists a unique tight set $\mathcal{T}^{(1)*}$, such that $|\mathcal{T}^{(1)*}| \leq |\mathcal{T}^{(1)}(x)|$ for all $x \in \mathcal{P}^{(1)*}$. Moreover $\mathcal{T}^{(1)*} \subseteq \mathcal{T}^{(1)}(x)$.

$\mathcal{T}^{(1)*} = \mathcal{T}^{(1)}(\nu)$ is called the *minimal tight set*.

Problem #2: how to find it?

Primal tight sets



Primal LP sequence

$$
\begin{aligned}
\min_{x^{(2)},\varepsilon^{(2)}} \quad & \varepsilon^{(2)} & & & & \\
\text{s.t.} \quad & \varepsilon^{(2)} + x^{(2)}(S) & \geq v(S) & & \forall S \subsetneq N, S \neq \emptyset & \qquad (\mathrm{P}^{(2)}) \\
& x^{(2)}(S) & = v(S) - w^*_S & & \forall S \in \mathcal{H}^{(1)} &
\end{aligned}
$$

where

- settled collection: $\mathcal{H}^{(1)} = N \cup \mathcal{T}^{(1)*}$
- at excess level: $w^*_S = \varepsilon^{(1)*}$ for all $S \in \mathcal{T}^{(1)*}$ and $w^*_N = 0$

Primal LP sequence

$$
\begin{aligned}
\min_{x^{(2)},\varepsilon^{(2)}} \quad & \varepsilon^{(2)} & & \\
\text{s.t.} \quad & \varepsilon^{(2)} + x^{(2)}(S) & \geq v(S) \qquad & \forall S \subsetneq N, S \neq \emptyset, S \notin \mathcal{T}^{(1)*} \\
& x^{(2)}(S) & = v(S) - w_S^* \qquad & \forall S \in \mathcal{H}^{(1)}
\end{aligned}
\tag{P$^{(2)}$}
$$

Primal LP sequence

$$
\begin{aligned}
\min_{x^{(2)},\varepsilon^{(2)}} \quad & \varepsilon^{(2)} & & \\
\text{s.t.} \quad & \varepsilon^{(2)} + x^{(2)}(S) & \geq v(S) \qquad & \forall S \subsetneq N, S \neq \emptyset, S \notin \mathcal{T}^{(1)*} \\
& x^{(2)}(S) & = v(S) - w_S^* \qquad & \forall S \in \mathcal{R}^{(1)}
\end{aligned}
\tag{P$^{(2)}$}
$$

where

- Representation:

$$\text{span}(\mathcal{R}^{(1)}) = \text{span}(\mathcal{H}^{(1)}), |\mathcal{R}^{(1)}| = \text{rank}(\mathcal{R}^{(1)})$$

Primal LP sequence

$$
\begin{aligned}
\min_{x^{(2)},\varepsilon^{(2)}} \quad & \varepsilon^{(2)} & & & & \\
\text{s.t.} \quad & \varepsilon^{(2)} + x^{(2)}(S) & \geq v(S) & & \forall S \notin \text{span}(\mathcal{R}^{(1)}) & \qquad (\mathrm{P}^{(2)}) \\
& x^{(2)}(S) & = v(S) - w_S^* & & \forall S \in \mathcal{R}^{(1)} &
\end{aligned}
$$

Primal LP sequence

$$
\begin{aligned}
\min_{x^{(k)},\varepsilon^{(k)}} \quad & \varepsilon^{(k)} & & & \\
\text{s.t.} \quad & \varepsilon^{(k)} + x^{(k)}(S) & \geq v(S) & \quad \forall S \notin \text{span}(\mathcal{R}^{(k-1)}) & \quad (\text{P}^{(k)}) \\
& x^{(k)}(S) & = v(S) - w_S^* & \quad \forall S \in \mathcal{R}^{(k-1)} &
\end{aligned}
$$

Primal LP sequence

$$
\begin{aligned}
\min_{x^{(k)},\varepsilon^{(k)}} \quad & \varepsilon^{(k)} & & & \\
\text{s.t.} \quad & \varepsilon^{(k)} + x^{(k)}(S) & \geq v(S) & \forall S \notin \text{span}(\mathcal{R}^{(k-1)}) & (\text{P}^{(k)}) \\
& x^{(k)}(S) & = v(S) - w_S^* & \forall S \in \mathcal{R}^{(k-1)} &
\end{aligned}
$$

where

- $\mathcal{T}^{(0)*} = N, \mathcal{H}^{(k)} = \bigcup_{i=0}^{k} \mathcal{T}^{(i)*}$,

Primal LP sequence

$$
\begin{aligned}
\min_{x^{(k)},\varepsilon^{(k)}} \quad & \varepsilon^{(k)} & & & \\
\text{s.t.} \quad & \varepsilon^{(k)} + x^{(k)}(S) & \geq v(S) & \forall S \notin \text{span}(\mathcal{R}^{(k-1)}) & (\text{P}^{(k)}) \\
& x^{(k)}(S) & = v(S) - w_S^* & \forall S \in \mathcal{R}^{(k-1)} &
\end{aligned}
$$

where

- $\mathcal{T}^{(0)*} = N, \mathcal{H}^{(k)} = \bigcup_{i=0}^{k} \mathcal{T}^{(i)*}$,
- $\text{span}(\mathcal{R}^{(k)}) = \text{span}(\mathcal{H}^{(k)}), |\mathcal{R}^{(k)}| = \text{rank}(\mathcal{R}^{(k)})$,

Primal LP sequence

$$
\begin{aligned}
\min_{x^{(k)},\varepsilon^{(k)}} \quad & \varepsilon^{(k)} & & & \\
\text{s.t.} \quad & \varepsilon^{(k)} + x^{(k)}(S) & \geq v(S) & \quad \forall S \notin \text{span}(\mathcal{R}^{(k-1)}) & \quad (\text{P}^{(k)}) \\
& x^{(k)}(S) & = v(S) - w_S^* & \quad \forall S \in \mathcal{R}^{(k-1)} &
\end{aligned}
$$

where

- $\mathcal{T}^{(0)*} = N, \mathcal{H}^{(k)} = \bigcup_{i=0}^{k} \mathcal{T}^{(i)*}$,
- $\text{span}(\mathcal{R}^{(k)}) = \text{span}(\mathcal{H}^{(k)}), |\mathcal{R}^{(k)}| = \text{rank}(\mathcal{R}^{(k)})$,
- $\forall k \geq 1 : w_S^* = \varepsilon^{(k)*}$ for all $S \in \mathcal{T}^{(k)*}$, and $w_N^* = 0$.

Primal LP sequence

$$
\begin{aligned}
\min_{x^{(k)},\varepsilon^{(k)}} \quad & \varepsilon^{(k)} & & & \\
\text{s.t.} \quad & \varepsilon^{(k)} + x^{(k)}(S) & \geq v(S) & \quad \forall S \notin \text{span}(\mathcal{R}^{(k-1)}) & \quad (\text{P}^{(k)}) \\
& x^{(k)}(S) & = v(S) - w_S^* & \quad \forall S \in \mathcal{R}^{(k-1)} &
\end{aligned}
$$

where

- $\mathcal{T}^{(0)*} = N, \mathcal{H}^{(k)} = \bigcup_{i=0}^{k} \mathcal{T}^{(i)*}$,
- $\text{span}(\mathcal{R}^{(k)}) = \text{span}(\mathcal{H}^{(k)}), |\mathcal{R}^{(k)}| = \text{rank}(\mathcal{R}^{(k)})$,
- $\forall k \geq 1 : w_S^* = \varepsilon^{(k)*}$ for all $S \in \mathcal{T}^{(k)*}$, and $w_N^* = 0$.
- We have to solve $(\text{P}^{(k)})$ at most $n-1$ times.

$$
n \geq \text{rank}(\mathcal{R}^{(k)}) \geq \text{rank}(\mathcal{R}^{(k-1)}) + 1 \geq k + \text{rank}(\mathcal{R}^{(0)}) = k + 1
$$

Primal LP sequence

$$
\begin{array}{lll}
\min\limits_{x^{(k)},\varepsilon^{(k)}} & \varepsilon^{(k)} & \\
\text{s.t.} \quad \varepsilon^{(k)} + x^{(k)}(S) & \geq v(S) & \forall S \notin \text{span}(\mathcal{R}^{(k-1)}) \\
x^{(k)}(S) & = v(S) - w_S^* & \forall S \in \mathcal{R}^{(k-1)}
\end{array}
\qquad (\mathrm{P}^{(k)})
$$

where

- $\mathcal{T}^{(0)*} = N, \mathcal{H}^{(k)} = \bigcup_{i=0}^{k} \mathcal{T}^{(i)*}$,
- $\text{span}(\mathcal{R}^{(k)}) = \text{span}(\mathcal{H}^{(k)}), |\mathcal{R}^{(k)}| = \text{rank}(\mathcal{R}^{(k)})$,
- $\forall k \geq 1 : w_S^* = \varepsilon^{(k)*}$ for all $S \in \mathcal{T}^{(k)*}$, and $w_N^* = 0$.
- We have to solve $(\mathrm{P}^{(k)})$ at most $n-1$ times.

$$n \geq \text{rank}(\mathcal{R}^{(k)}) \geq \text{rank}(\mathcal{R}^{(k-1)}) + 1 \geq k + \text{rank}(\mathcal{R}^{(0)}) = k + 1$$

- Problem #3: how to find $\text{span}(\mathcal{R}^{(k)})$?

Dual LP sequence

$$
\begin{aligned}
\max_{u^{(k)},z^{(k)}} \quad & \sum_{S \notin \text{span}(\mathcal{R}^{(k-1)})} u_S^{(k)} v(S) \quad + \sum_{S \in \mathcal{R}^{(k-1)}} z_S^{(k)} (v(S) - w_S^*) \\
\text{s.t.} \quad & \sum_{S \notin \text{span}(\mathcal{R}^{(k-1)})} u_S^{(k)} e(S)_i \quad + \sum_{S \in \mathcal{R}^{(k-1)}} z_S^{(k)} e(S)_i = 0 \quad \forall i \in N \\
& \sum_{S \notin \text{span}(\mathcal{R}^{(k-1)})} u_S^{(k)} \quad = 1 \\
& u^{(k)} \quad \geq 0
\end{aligned}
\tag{D$^{(k)}$}
$$

Dual LP sequence

$$
\begin{aligned}
\max_{u^{(k)}, z^{(k)}} \quad & \sum_{S \notin \text{span}(\mathcal{R}^{(k-1)})} u_S^{(k)} v(S) \quad + \sum_{S \in \mathcal{R}^{(k-1)}} z_S^{(k)} (v(S) - w_S^*) \\
\text{s.t.} \quad & \sum_{S \notin \text{span}(\mathcal{R}^{(k-1)})} u_S^{(k)} e(S)_i \quad + \sum_{S \in \mathcal{R}^{(k-1)}} z_S^{(k)} e(S)_i = 0 \quad \forall i \in N \\
& \sum_{S \notin \text{span}(\mathcal{R}^{(k-1)})} u_S^{(k)} \qquad = 1 \\
& u^{(k)} \qquad \geq 0
\end{aligned}
\tag{D$^{(k)}$}
$$

For dual feasible $u^{(k)} \in \mathcal{D}^{(k)}$, the *dual non-tight set* is

$$\mathcal{DNT}^{(k)}(u^{(k)}) = \{S \notin \text{span}(\mathcal{R}^{(k-1)}) : u_S^{(k)} > 0\}$$

Theorem 1 (maximal non-tight set)

For any primal-dual optimal pair $x^{(k)} \in \mathcal{P}^{(k)*}$, $u^{(k)} \in \mathcal{D}^{(k)*}$

$$\mathcal{DNT}^{(k)}(u^{(k)}) \subseteq \mathcal{DNT}^{(k)*} = \mathcal{T}^{(k)*} \subseteq \mathcal{T}^{(k)}(x^{(k)})$$

Sequential LP algorithm framework

Algorithm 1: Finding the nucleolus ν of cooperative game v

Input: (N, v);

1. Initialisation: $k = 1, \mathcal{R}^{(0)} = N, w_N^* = 0$;

while rank$(\mathcal{R}^{(k-1)}) < n$ **do**

2. Solve $(\mathrm{P}^{(k)})$ (or $(\mathrm{D}^{(k)})$): finding $(x^{(k)}, \epsilon^{(k)*}) \in \mathcal{P}^{(k)*}$ (or $(u^{(k)}, z^{(k)}) \in \mathcal{D}^{(k)*}$);

3. Find (the entire, or some subset of) the minimal tight set: $\mathcal{T} \subseteq \mathcal{T}^{(k)*}$;

4. Update $\mathcal{R}^{(k-1)}$: span $\left(\mathcal{R}^{(k)}\right) =$ span $\left(\mathcal{R}^{(k-1)} \cup \mathcal{T}\right)$, rank $\left(\mathcal{R}^{(k)}\right) = |\mathcal{R}^{(k)}|$, $k = k + 1$;

5. Find all $S \notin$ span $\left(\mathcal{R}^{(k-1)}\right)$ to formulate $(\mathrm{P}^{(k)})$ (or $(\mathrm{D}^{(k)})$);

end

Output: ν is the unique solution of $x(S) = v(S) - w_S^* \ \forall S \in \mathcal{R}^{(k-1)}$;

Sequential LP algorithm framework

Algorithm 2: Finding the nucleolus ν of cooperative game v

Input: (N, v);

1. Initialisation: $k = 1, \mathcal{R}^{(0)} = N, w_N^* = 0$;

while rank$(\mathcal{R}^{(k-1)}) < n$ **do**

2. Solve $(\mathrm{P}^{(k)})$ (or $(\mathrm{D}^{(k)})$): finding $(x^{(k)}, \epsilon^{(k)*}) \in \mathcal{P}^{(k)*}$ (or $(u^{(k)}, z^{(k)}) \in \mathcal{D}^{(k)*}$);

3. Find (the entire, or some subset of) the minimal tight set: $\mathcal{T} \subseteq \mathcal{T}^{(k)*}$;

4. Update $\mathcal{R}^{(k-1)}$: span $\left(\mathcal{R}^{(k)}\right) =$ span $\left(\mathcal{R}^{(k-1)} \cup \mathcal{T}\right)$, rank $\left(\mathcal{R}^{(k)}\right) = |\mathcal{R}^{(k)}|$, $k = k + 1$;

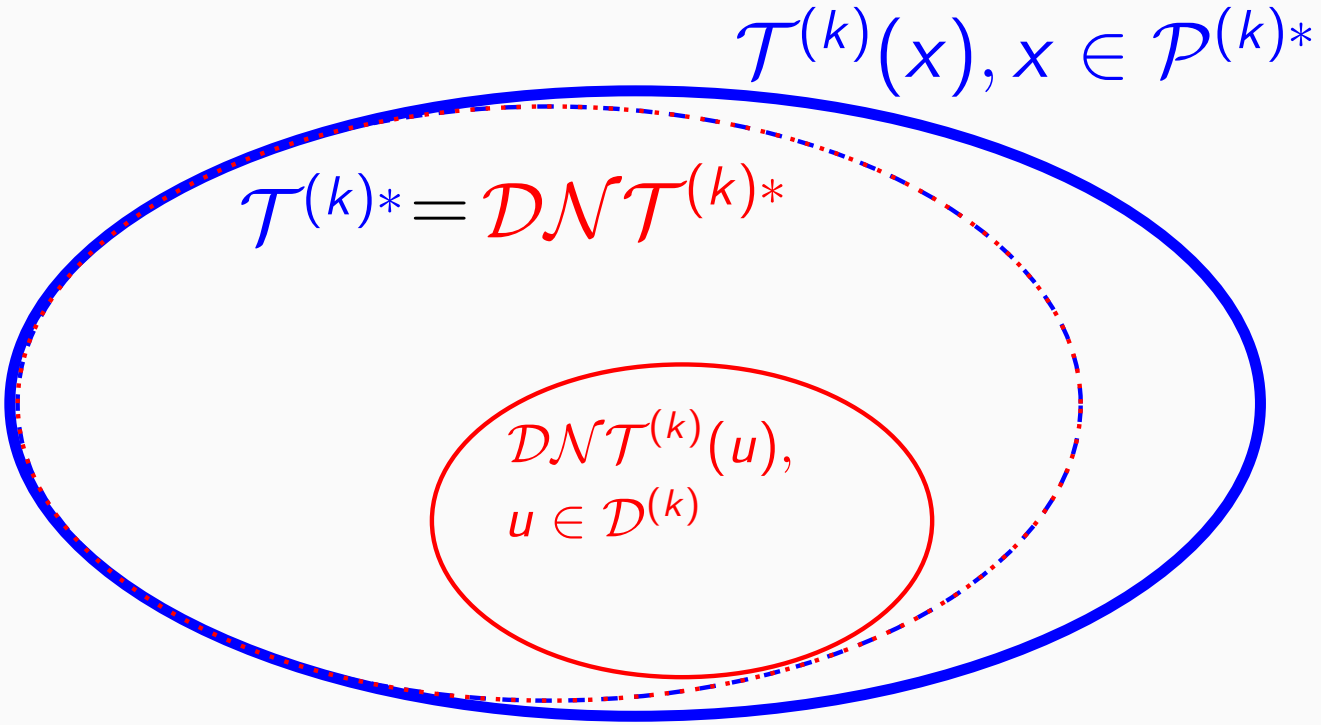
5. Find all $S \notin$ span $\left(\mathcal{R}^{(k-1)}\right)$ to formulate $(\mathrm{P}^{(k)})$ (or $(\mathrm{D}^{(k)})$);

end

Output: ν is the unique solution of $x(S) = v(S) - w_S^* \ \forall S \in \mathcal{R}^{(k-1)}$;

Problems #1, #2 and #3 correspond to performing **Steps 2, 3** and **5**

Primal-dual tight sets



Minimal tight set

Finding the minimal tight set

Assuming we can handle Problem #1, focus on
Problem #2: how do we find the minimal tight set?

Finding the minimal tight set

Assuming we can handle Problem #1, focus on
Problem #2: how do we find the minimal tight set?

1. Interior point methods (IPM)
    - Solving the large Newton system with (dense) $AA^\top$ is prohibitive;

Finding the minimal tight set

Assuming we can handle Problem #1, focus on
Problem #2: how do we find the minimal tight set?

1. Interior point methods (IPM)
    - Solving the large Newton system with (dense) $AA^\top$ is prohibitive; formulating the small (and also dense) dual $A^\top A$ similarly challenging.

Finding the minimal tight set

Assuming we can handle Problem #1, focus on
Problem #2: how do we find the minimal tight set?

1. Interior point methods (IPM)
2. Strict complementary solutions without IPMs: [Freund et al., 1985]
   - Moving from $x^{(k)}$ to $x^{(k)*}$ with $\mathcal{T}^{(k)}(x^{(k)*}) = \mathcal{T}^{(k)*}$. Requires solving a larger LP once.

Finding the minimal tight set

Assuming we can handle Problem #1, focus on
Problem #2: how do we find the minimal tight set?

1. Interior point methods (IPM)
2. Strict complementary solutions without IPMs
3. Improving subroutine (*FBOS*) of [Nguyen and Thomas, 2016]
   - Moving from $x^{(k)}$ to $\tilde{x}^{(k)}$ with $|\mathcal{T}^{(k)}(\tilde{x}^{(k)})| < |\mathcal{T}^{(k)}(x^{(k)})|$. Only requires solving a sequence of small LPs, but numerically sensitive (Lemma 1 in practice).

Finding the minimal tight set

Assuming we can handle Problem #1, focus on
Problem #2: how do we find the minimal tight set?

1. Interior point methods (IPM)
2. Strict complementary solutions without IPMs
3. Improving primal subroutine (*FBOS*)
4. Dual solution
    - Theorem 1: every dual non-tight set is within the minimal tight set.

Finding the minimal tight set

Assuming we can handle Problem #1, focus on
Problem #2: how do we find the minimal tight set?

1. Interior point methods (IPM)
2. Strict complementary solutions without IPMs
3. Improving primal subroutine (*FBOS*)
4. Dual solution
    - Theorem 1: every dual non-tight set is within the minimal tight set. Moreover, the maximal dual non-tight set coincides with the minimal tight set.

Recall the dual LP sequence and Theorem 1

$$
\begin{aligned}
\max_{u^{(k)},z^{(k)}} \quad & \sum_{S\notin \mathrm{span}(\mathcal{R}^{(k-1)})} u_S^{(k)} v(S) \quad + \sum_{S\in\mathcal{R}^{(k-1)}} z_S^{(k)}(v(S)-w_S^*) \\
\text{s.t.} \quad & \sum_{S\notin \mathrm{span}(\mathcal{R}^{(k-1)})} u_S^{(k)} e(S)_i \quad + \sum_{S\in\mathcal{R}^{(k-1)}} z_S^{(k)} e(S)_i = 0 \quad \forall i \in N \\
& \sum_{S\notin \mathrm{span}(\mathcal{R}^{(k-1)})} u_S^{(k)} \quad = 1 \\
& u^{(k)} \quad \geq 0
\end{aligned}
\tag{$\mathrm{D}^{(k)}$}
$$

$$
\mathcal{DNT}^{(k)}(u^{(k)}) \subseteq \mathcal{DNT}^{(k)*} = \mathcal{T}^{(k)*} \subseteq \mathcal{T}^{(k)}(x^{(k)})
$$

Dual algorithms ([Solymosi, 1993], [Derks and Kuipers, 1997])

Step 2: Solve ($\mathrm{D}^{(k)}$): find any optimal $u^{(k)} \in \mathcal{D}^{(k)*}$; **Step 3:** -;

Step 4: Update $\mathcal{R}^{(k-1)}$: $\mathrm{rank}\left(\mathcal{R}^{(k)}\right) = \mathrm{rank}\left(\mathcal{R}^{(k-1)} \cup \{S : u_S^{(k)} > 0\}\right)$;

Recall the dual LP sequence and Theorem 1

$$
\begin{aligned}
\max_{u^{(k)},z^{(k)}} \quad & \sum_{S\notin \mathrm{span}(\mathcal{R}^{(k-1)})} u_S^{(k)} v(S) \quad + \sum_{S\in\mathcal{R}^{(k-1)}} z_S^{(k)}(v(S)-w_S^*) \\
\text{s.t.} \quad & \sum_{S\notin \mathrm{span}(\mathcal{R}^{(k-1)})} u_S^{(k)} e(S)_i \quad + \sum_{S\in\mathcal{R}^{(k-1)}} z_S^{(k)} e(S)_i = 0 \quad \forall i \in N \\
& \sum_{S\notin \mathrm{span}(\mathcal{R}^{(k-1)})} u_S^{(k)} \quad = 1 \\
& u^{(k)} \quad \geq 0
\end{aligned}
\tag{$\mathrm{D}^{(k)}$}
$$

$$
\mathcal{DNT}^{(k)}(u^{(k)}) \subseteq \mathcal{DNT}^{(k)*} = \mathcal{T}^{(k)*} \subseteq \mathcal{T}^{(k)}(x^{(k)})
$$

Dual algorithms ([Solymosi, 1993], [Derks and Kuipers, 1997])

Step 2: Solve ($\mathrm{D}^{(k)}$): find any optimal $u^{(k)} \in \mathcal{D}^{(k)*}$; **Step 3:** -;

Step 4: Update $\mathcal{R}^{(k-1)}$: $\mathrm{rank}\left(\mathcal{R}^{(k)}\right) = \mathrm{rank}\left(\mathcal{R}^{(k-1)} \cup \{S : u_S^{(k)} > 0\}\right)$;

Primal-dual trade-off: number of iterations vs. compact bases!

Support of dual feasible solutions

Optimality is overrated compared to feasibility.

Primal algorithms: given $x^{(k)} \in \mathcal{P}^{(k)*}$ and $\mathcal{T}^{(k)} := \mathcal{T}^{(k)}(x^{(k)})$, repeatedly solve the **relaxed dual feasibility** problem

$$\begin{aligned}
\sum_{S \in \mathcal{T}^{(k)}} u_S^{(k)} e(S)_i + \sum_{S \in \mathcal{R}^{(k-1)}} z_S^{(k)} e(S)_i &= 0 \quad \forall i \in N \\
\sum_{S \in \mathcal{T}^{(k)}} u_S^{(k)} &= 1 \qquad \text{(subr)} \\
u^{(k)} &\geq 0
\end{aligned}$$

Support of dual feasible solutions

Optimality is overrated compared to feasibility.

Primal algorithms: given $x^{(k)} \in \mathcal{P}^{(k)*}$ and $\mathcal{T}^{(k)} := \mathcal{T}^{(k)}(x^{(k)})$, repeatedly solve the **relaxed dual feasibility** problem

$$\begin{aligned} \sum_{S \in \mathcal{T}^{(k)}} u_S^{(k)} e(S)_i + \sum_{S \in \mathcal{R}^{(k-1)}} z_S^{(k)} e(S)_i &= 0 \quad \forall i \in N \\ \sum_{S \in \mathcal{T}^{(k)}} u_S^{(k)} &= 1 \qquad \text{(subr)} \\ u^{(k)} &\geq 0 \end{aligned}$$

- If feasible, we find $u^{(k)*}$ solution and let $U := \{S \in \mathcal{T}^{(k)} : u_S^{(k)*} > 0\}$

Support of dual feasible solutions

Optimality is overrated compared to feasibility.

Primal algorithms: given $x^{(k)} \in \mathcal{P}^{(k)*}$ and $\mathcal{T}^{(k)} := \mathcal{T}^{(k)}(x^{(k)})$, repeatedly solve the **relaxed dual feasibility** problem

$$\begin{aligned} \sum_{S \in \mathcal{T}^{(k)}} u_S^{(k)} e(S)_i + \sum_{S \in \mathcal{R}^{(k-1)}} z_S^{(k)} e(S)_i &= 0 \quad \forall i \in N \\ \sum_{S \in \mathcal{T}^{(k)}} u_S^{(k)} &= 1 \qquad \text{(subr)} \\ u^{(k)} &\geq 0 \end{aligned}$$

- If feasible, we find $u^{(k)*}$ solution and let $U := \{S \in \mathcal{T}^{(k)} : u_S^{(k)*} > 0\}$
- Then $U^* := \{S \in \mathcal{T}^{(k)} : S \in \text{span}(U \cup \mathcal{R}^{(k-1)})\}$

Support of dual feasible solutions

Optimality is overrated compared to feasibility.

Primal algorithms: given $x^{(k)} \in \mathcal{P}^{(k)*}$ and $\mathcal{T}^{(k)} := \mathcal{T}^{(k)}(x^{(k)})$, repeatedly solve the **relaxed dual feasibility** problem

$$\begin{aligned} \sum_{S \in \mathcal{T}^{(k)}} u_S^{(k)} e(S)_i + \sum_{S \in \mathcal{R}^{(k-1)}} z_S^{(k)} e(S)_i &= 0 \quad \forall i \in N \\ \sum_{S \in \mathcal{T}^{(k)}} u_S^{(k)} &= 1 \qquad \text{(subr)} \\ u^{(k)} &\geq 0 \end{aligned}$$

- If feasible, we find $u^{(k)*}$ solution and let $U := \{S \in \mathcal{T}^{(k)} : u_S^{(k)*} > 0\}$
- Then $U^* := \{S \in \mathcal{T}^{(k)} : S \in \text{span}(U \cup \mathcal{R}^{(k-1)})\}$
- Remove $\mathcal{T}^{(k)} \setminus U^*$, expand $\mathcal{R}^{(k-1)} \leftarrow U^*$ and solve again

Support of dual feasible solutions

Optimality is overrated compared to feasibility.

Primal algorithms: given $x^{(k)} \in \mathcal{P}^{(k)*}$ and $\mathcal{T}^{(k)} := \mathcal{T}^{(k)}(x^{(k)})$, repeatedly solve the **relaxed dual feasibility** problem

$$\begin{aligned} \sum_{S \in \mathcal{T}^{(k)}} u_S^{(k)} e(S)_i + \sum_{S \in \mathcal{R}^{(k-1)}} z_S^{(k)} e(S)_i &= 0 \quad \forall i \in N \\ \sum_{S \in \mathcal{T}^{(k)}} u_S^{(k)} &= 1 \qquad \text{(subr)} \\ u^{(k)} &\geq 0 \end{aligned}$$

- If feasible, we find $u^{(k)*}$ solution and let $U := \{S \in \mathcal{T}^{(k)} : u_S^{(k)*} > 0\}$
- Then $U^* := \{S \in \mathcal{T}^{(k)} : S \in \text{span}(U \cup \mathcal{R}^{(k-1)})\}$
- Remove $\mathcal{T}^{(k)} \setminus U^*$, expand $\mathcal{R}^{(k-1)} \leftarrow U^*$ and solve again
- If infeasible, we found $\mathcal{T}^{(k)*}$ (the union of all U^* throughout)

Solving the large LPs

Descent algorithm

Descent algorithm framework: iteratively perform the following steps

- Optimality test
- Improving direction
- Step size

Descent algorithm

Descent algorithm framework: iteratively perform the following steps

- Optimality test
- Improving direction
- Step size

Tackle Problems #1 and #2 at the same time with (subr).

Descent algorithm

Descent algorithm framework: iteratively perform the following steps

- Optimality test
- Improving direction
- Step size

Tackle Problems #1 and #2 at the same time with (subr).

For any $x \in ir(v)$, let

$$\varepsilon(x) := \max_{S \not\in \text{span}\left(\mathcal{R}^{(k-1)}\right)} E(S, x) \tag{1}$$

Then $(x, \varepsilon(x)) \in \mathcal{P}^{(k)}$ is primal feasible.

Descent algorithm

Descent algorithm framework: iteratively perform the following steps

- Optimality test
- Improving direction
- Step size

Tackle Problems #1 and #2 at the same time with (subr).

For any $x \in ir(v)$, let

$$\varepsilon(x) := \max_{S \nsubseteq \text{span}\left(\mathcal{R}^{(k-1)}\right)} E(S, x) \tag{1}$$

Then $(x, \varepsilon(x)) \in \mathcal{P}^{(k)}$ is primal feasible.

Run the iterative dual subroutine with $\mathcal{T}^{(k)}(x, \varepsilon(x))$.
Optimality test: (subr) is feasible $\iff$ $(x, \varepsilon(x)) \in \mathcal{P}^{(k)*}$

Improving directions

Optimality test: if (subr) is infeasible

Improving directions

Optimality test: if (subr) is infeasible $\Rightarrow$ Farkas' lemma: $\exists d \in \mathbb{R}^n$

$$\begin{aligned} d(S) &\geq 1 \quad \forall S \in \mathcal{T}^{(k)} \\ d(S) &= 0 \quad \forall S \in \mathcal{R}^{(k-1)} \end{aligned} \tag{impr-dir}$$

Improving directions: free to choose objective function, for example

$$\min_d \sum_{S \in \mathcal{T}^{(k)}} d(S)$$

Improving directions and optimal step size

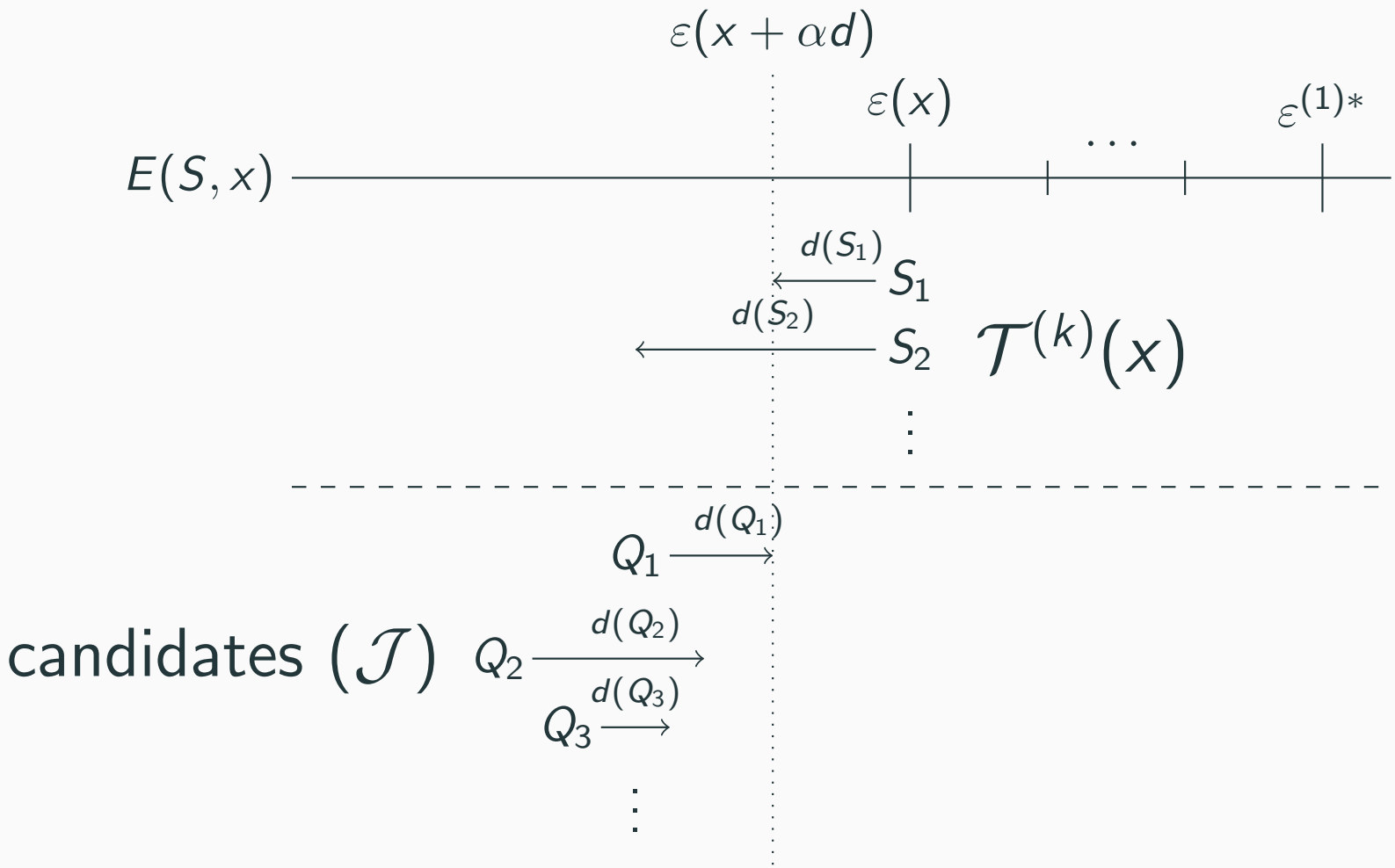


Figure 1: Optimal step size

Improving directions and optimal step size

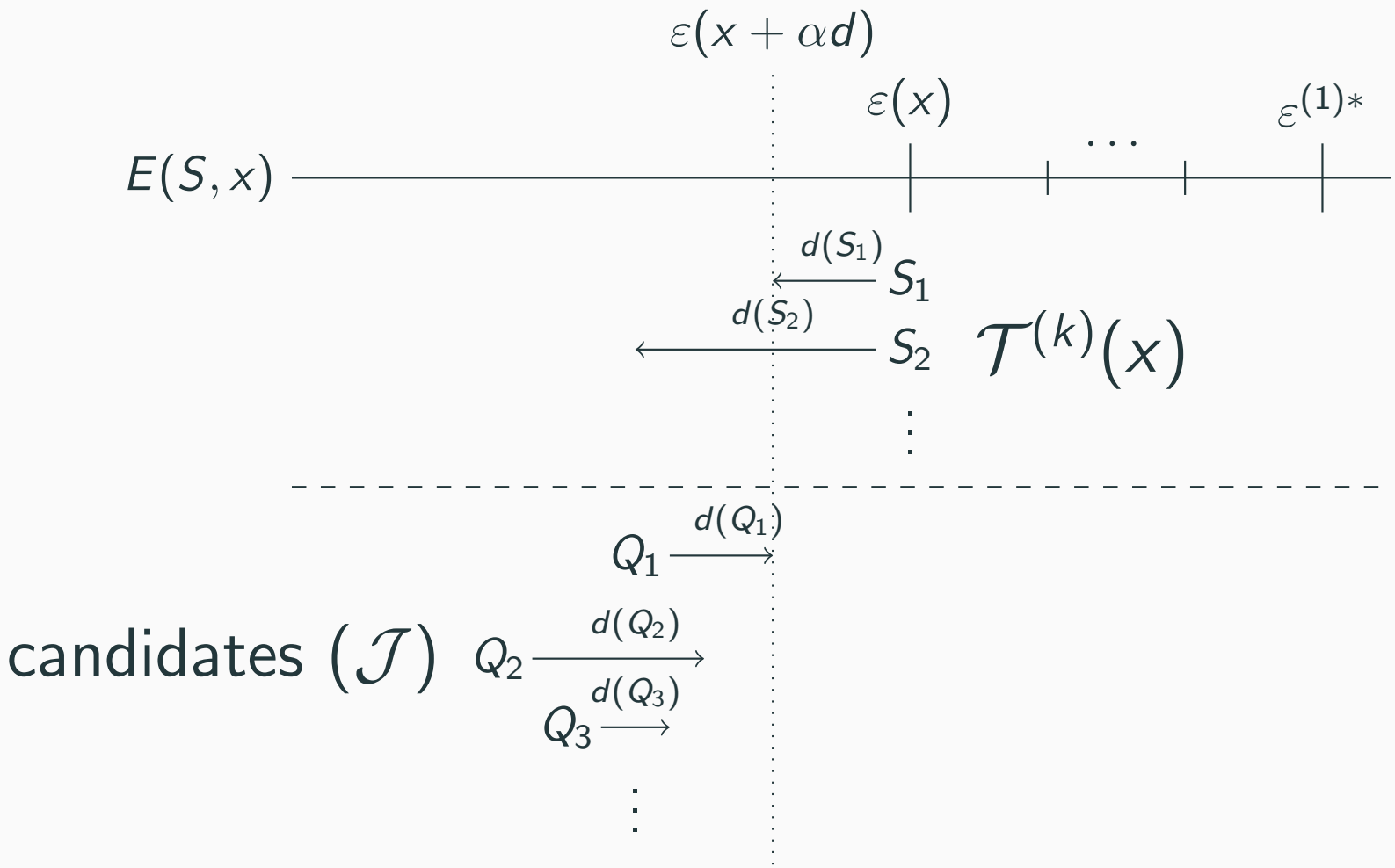


Figure 1: Optimal step size

The smallest step size such that $\mathcal{T}^{(k)}(x) \neq \mathcal{T}^{(k)}(x+\alpha d)$.

Lexicographical descent algorithm

Algorithm 3: Lexicographical descent algorithm computing the nucleolus

Input: (N, v);

1. Initialisation: $k = 1, \mathcal{R}^{(0)} = N$ and $x \in ir(v)$ arbitrary;

while $\text{rank}(\mathcal{R}^{(k-1)}) < n$ **do**

  2. Find $\varepsilon(x)$ with (1) and $\mathcal{T}^{(k)}(x, \varepsilon(x))$;

  if $\mathcal{T}^{(k)}(x, \varepsilon(x)) = \mathcal{T}^{(k)*}$ **then**

    3. Update $\mathcal{R}^{(k-1)}$: $\text{span}\left(\mathcal{R}^{(k)}\right) = \text{span}\left(\mathcal{R}^{(k-1)} \cup \mathcal{T}^{(k)*}\right)$, $\text{rank}\left(\mathcal{R}^{(k)}\right) = |\mathcal{R}^{(k)}|$, $k = k + 1$;

    4. Find all $S \notin \text{span}\left(\mathcal{R}^{(k-1)}\right)$;

  else

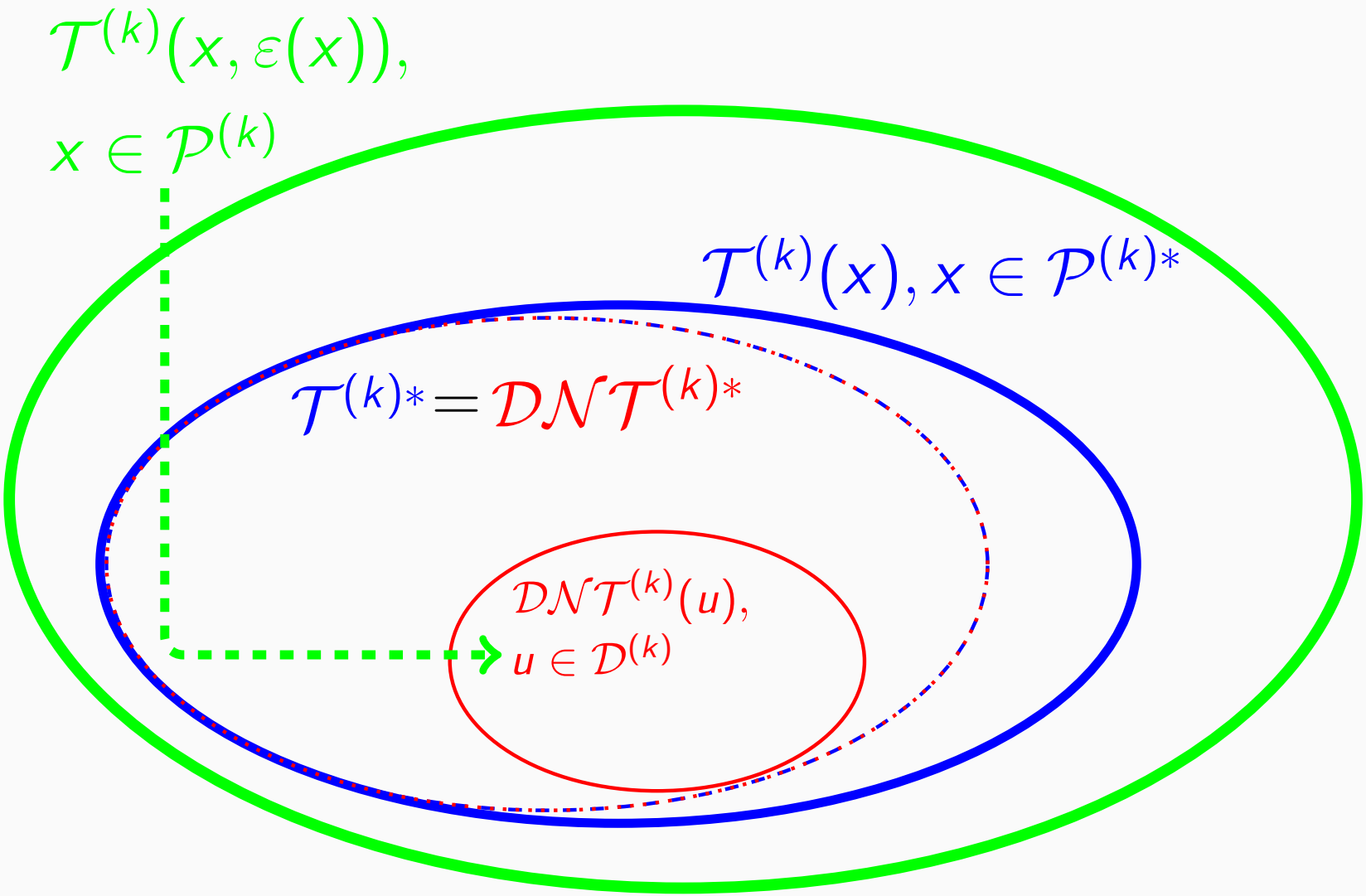
    5. Find d with (impr-dir), α (Fig. 1), make the step $x = x + \alpha d$ and go to **Step 2**;

  end

end

Output: $x = \nu$ is the nucleolus;

Primal-dual tight sets



Geometry

Problem

Informally: given a *special* polytope P

Problem

Informally: given a *special* polytope P

1. find the set of points $C \subseteq P$ that are centers of the largest balls within P

Problem

Informally: given a *special* polytope P

1. find the set of points $C \subseteq P$ that are centers of the largest balls within P
2. remove from P the edges limiting the size of the largest balls

Problem

Informally: given a *special* polytope P

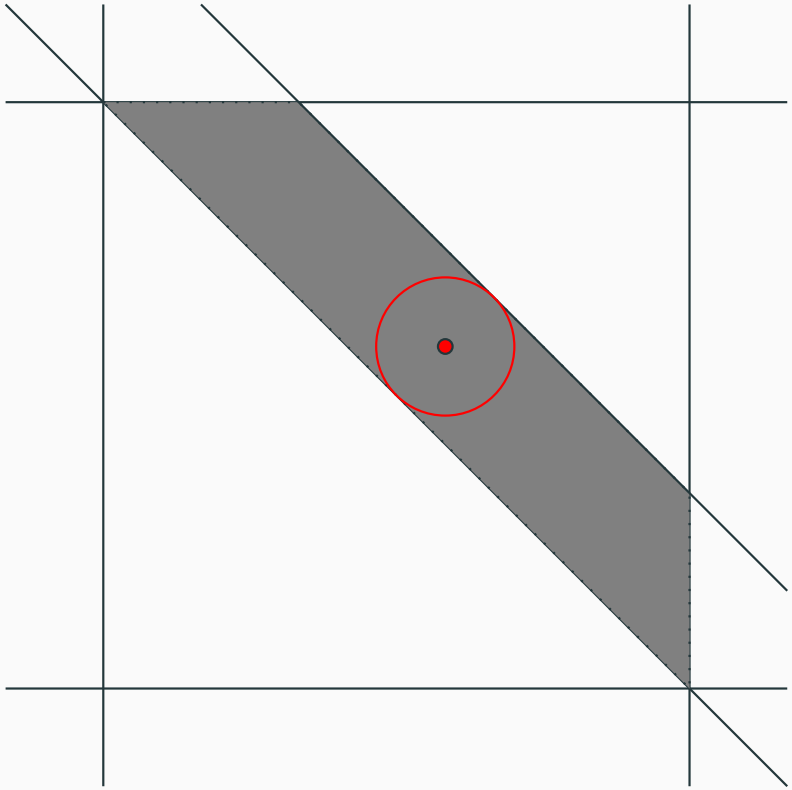
1. find the set of points $C \subseteq P$ that are centers of the largest balls within P
2. remove from P the edges limiting the size of the largest balls
3. remove from C the points that are not centers of the largest balls within P

Problem

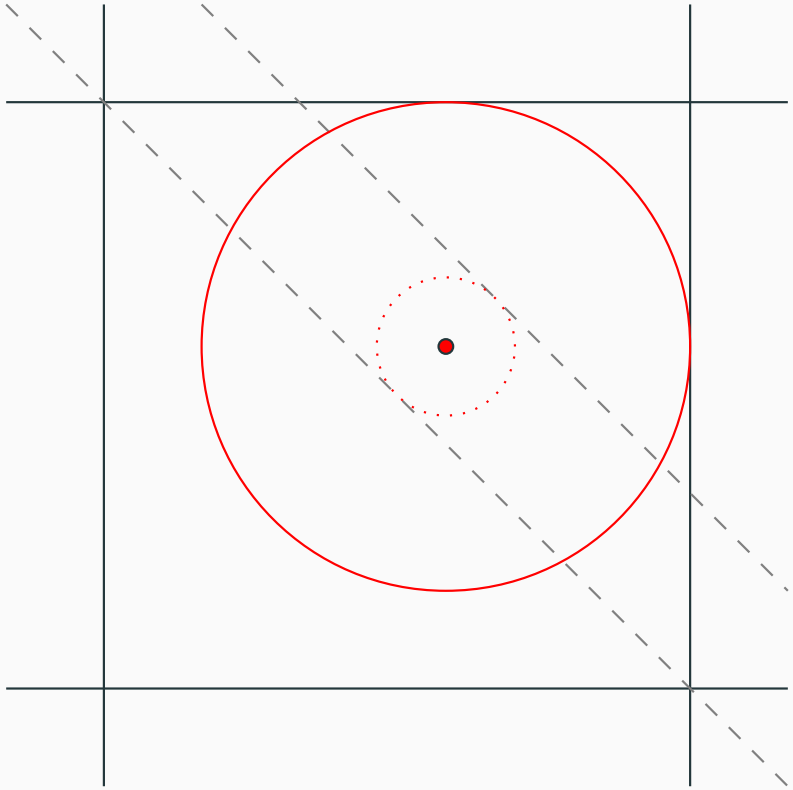
Informally: given a *special* polytope P

1. find the set of points $C \subseteq P$ that are centers of the largest balls within P
2. remove from P the edges limiting the size of the largest balls
3. remove from C the points that are not centers of the largest balls within P
4. repeat 2. and 3. until $|C| = 1$

Problem

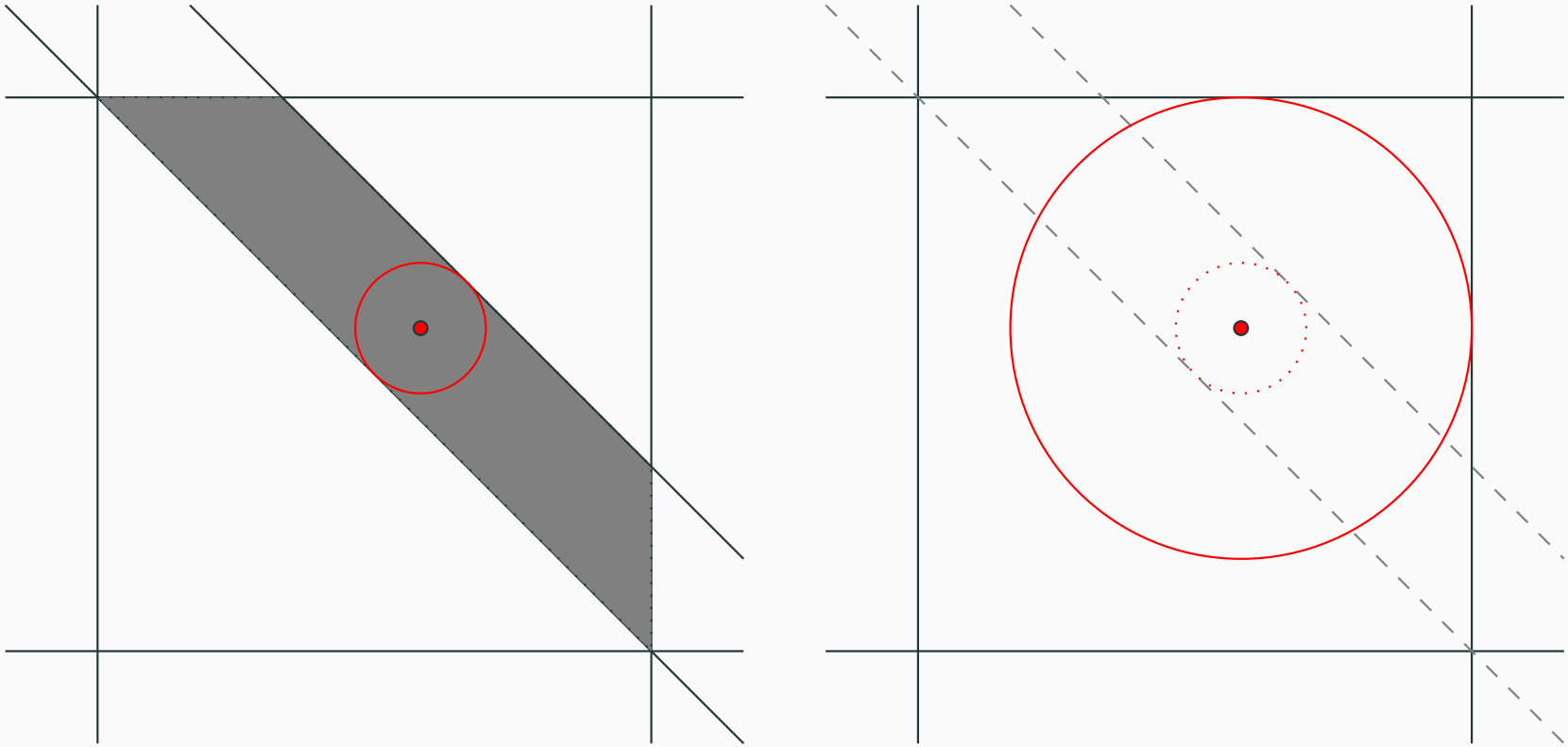


(a) Starting level



(b) Subsequent level

Problem



(a) Starting level

(b) Subsequent level

Figure 2: Geometric view of the nucleolus

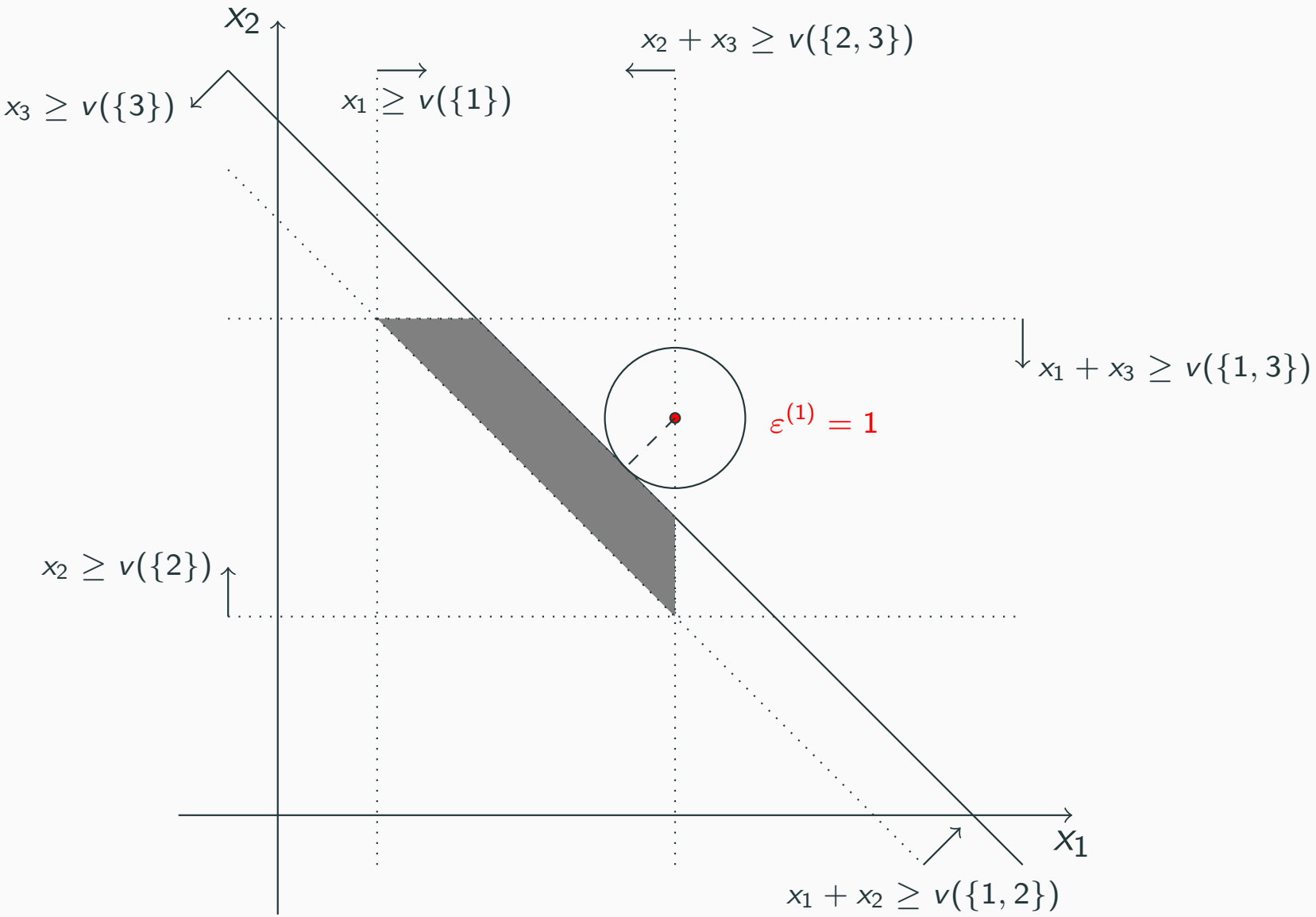


Figure 3: Hyperplane of allocations: $x_3 = v(N) - x_1 - x_2$

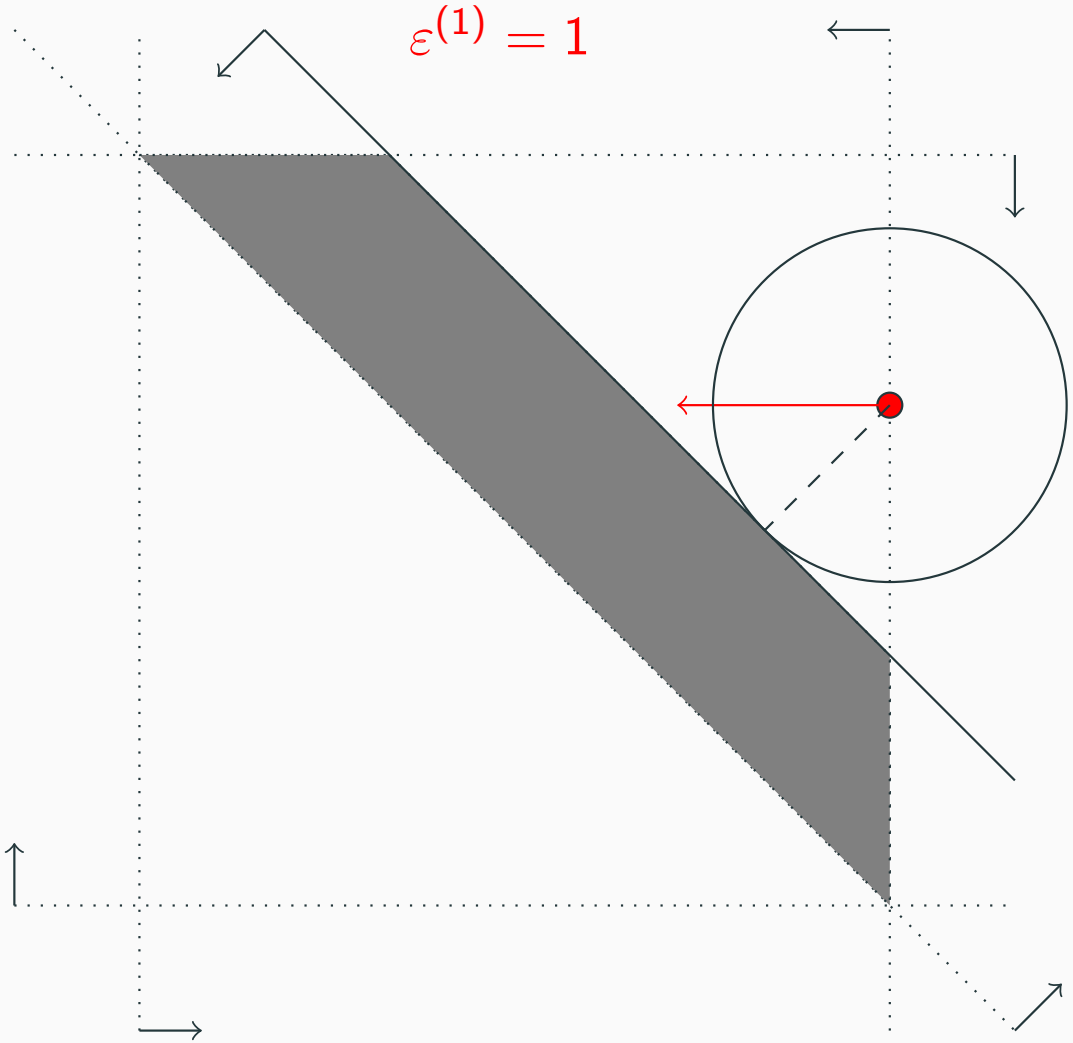


Figure 3: Improving direction

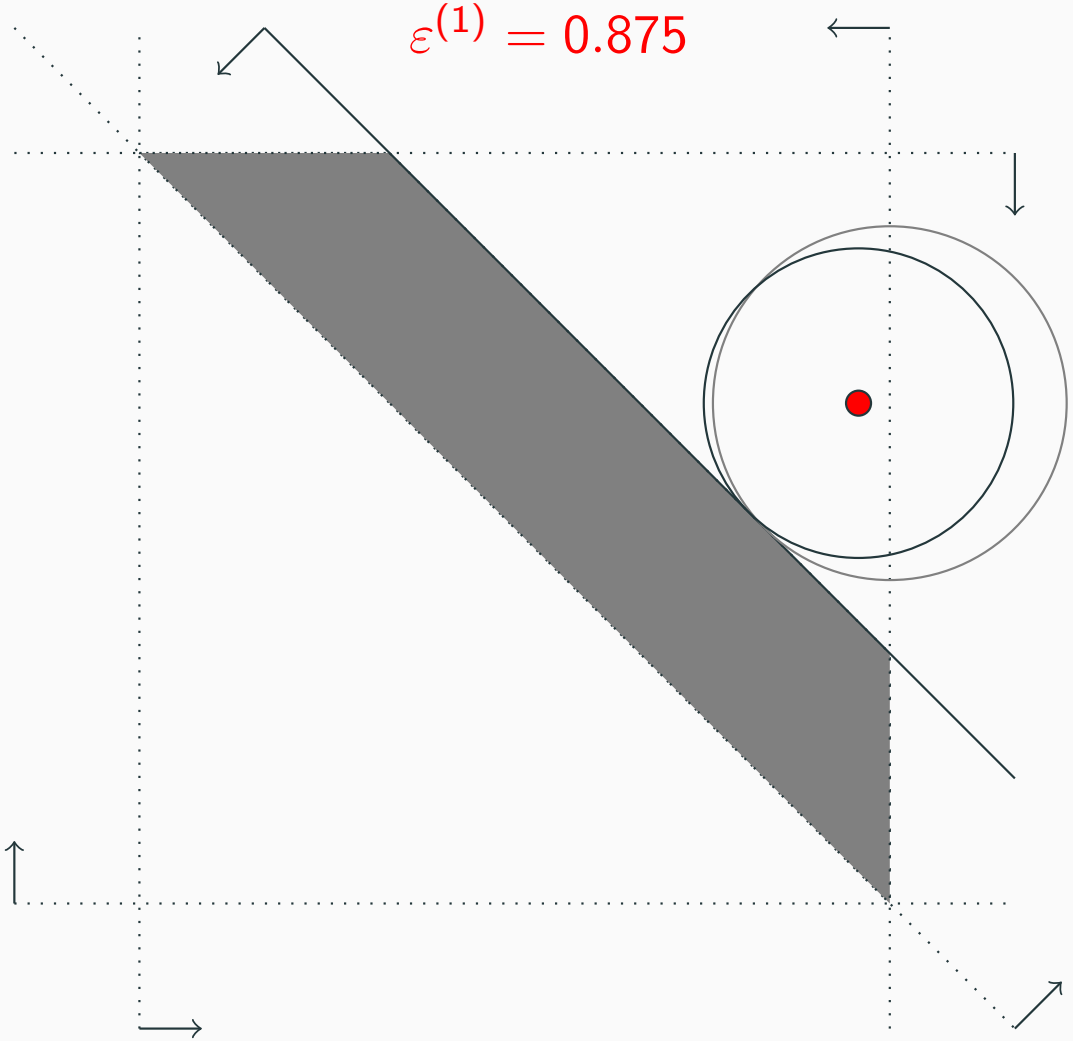


Figure 3: Pivot step

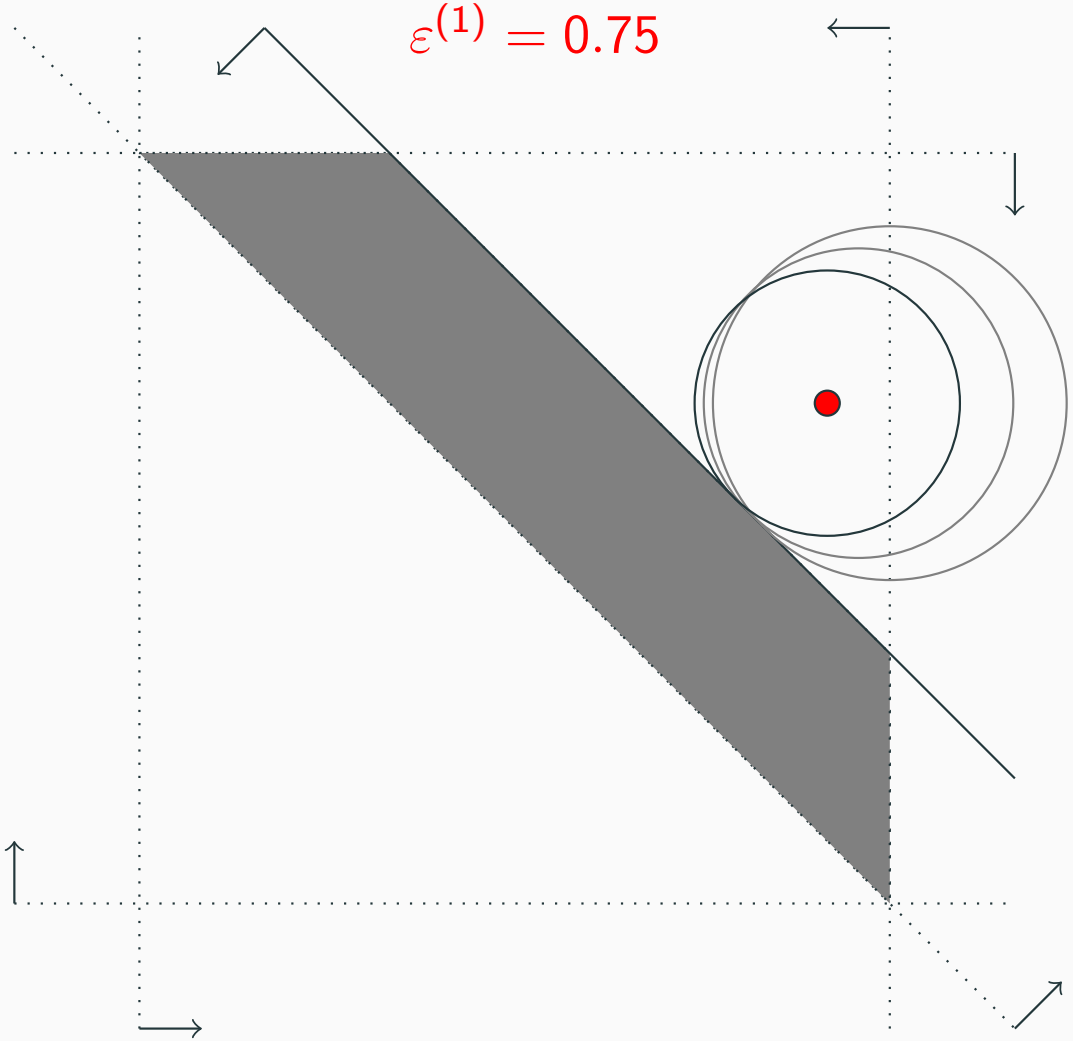


Figure 3: Pivot step

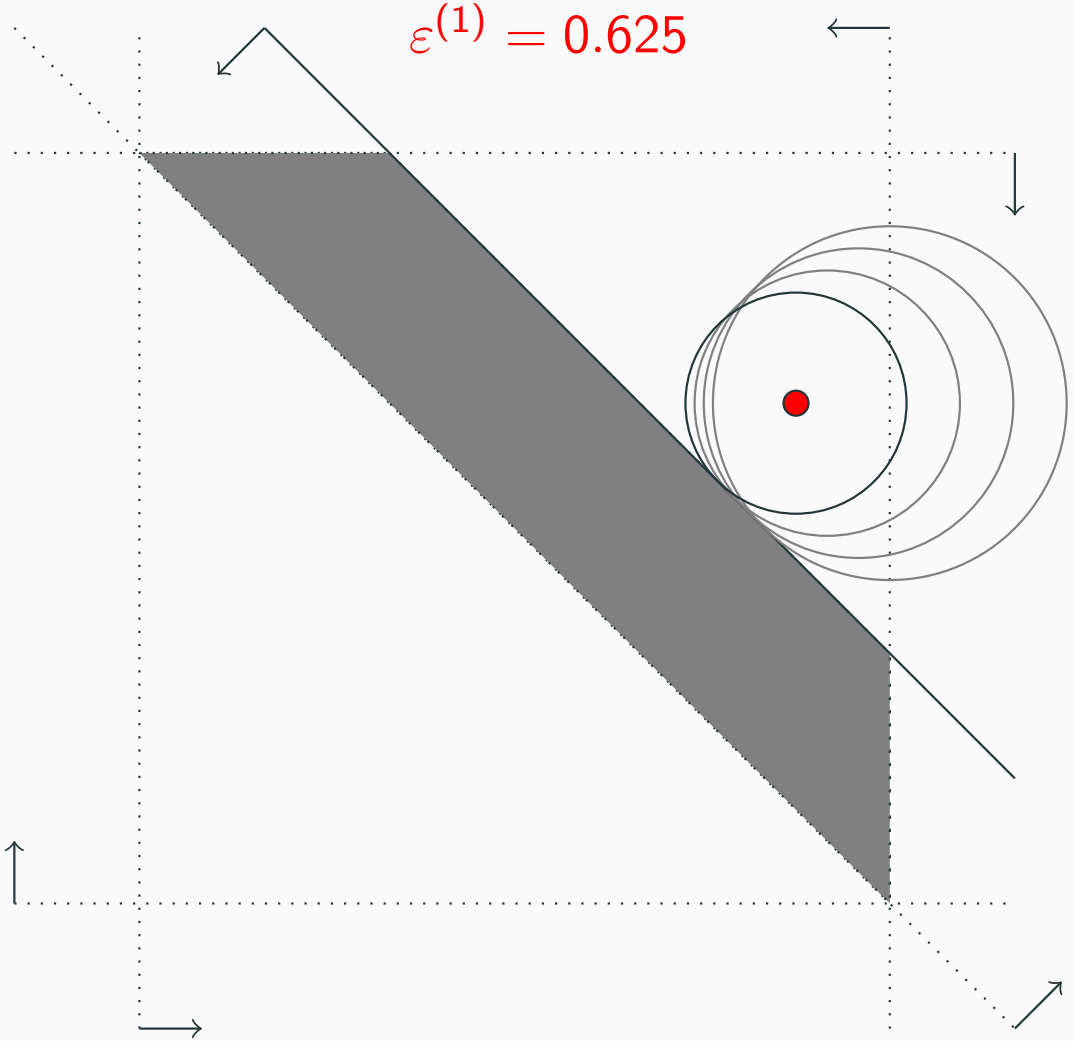


Figure 3: Pivot step

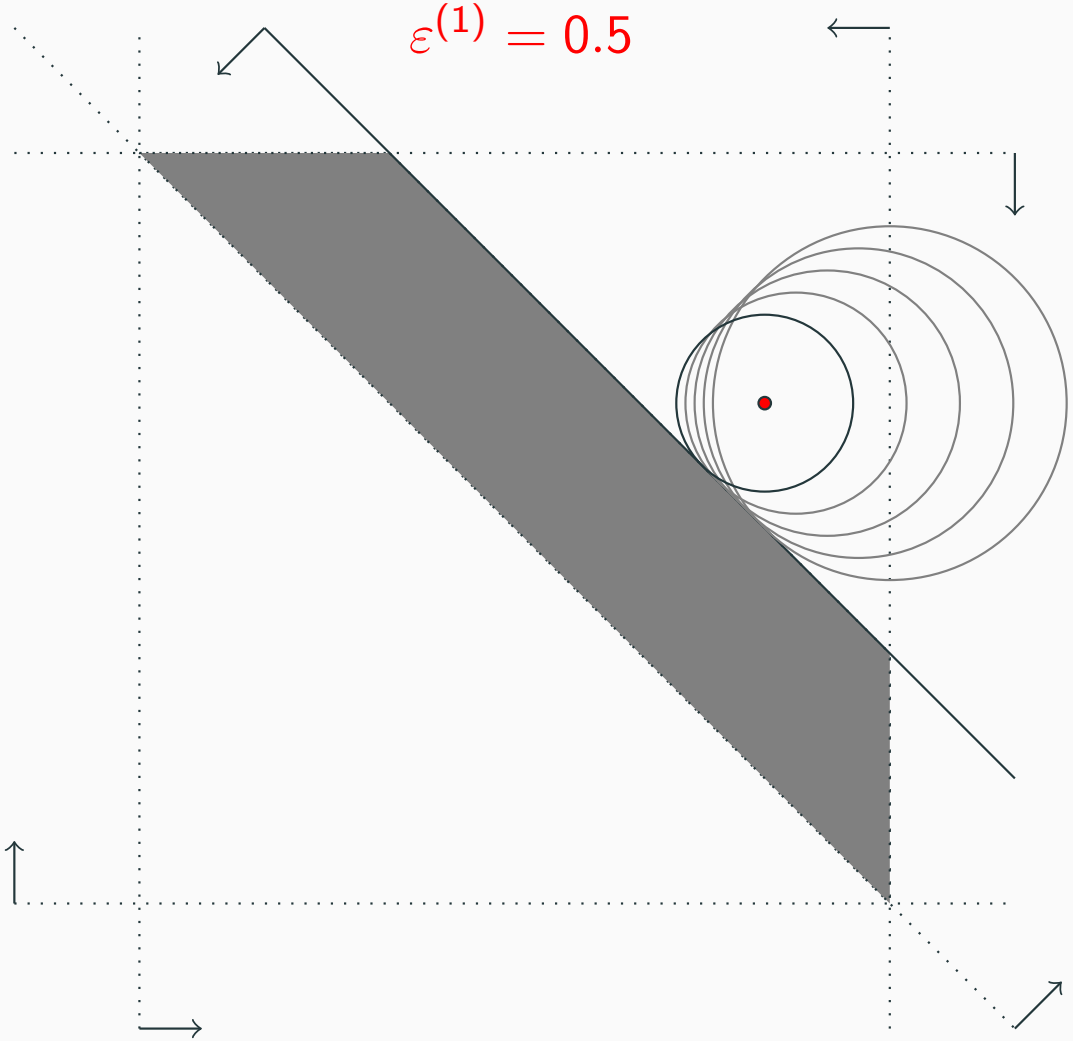


Figure 3: Pivot step

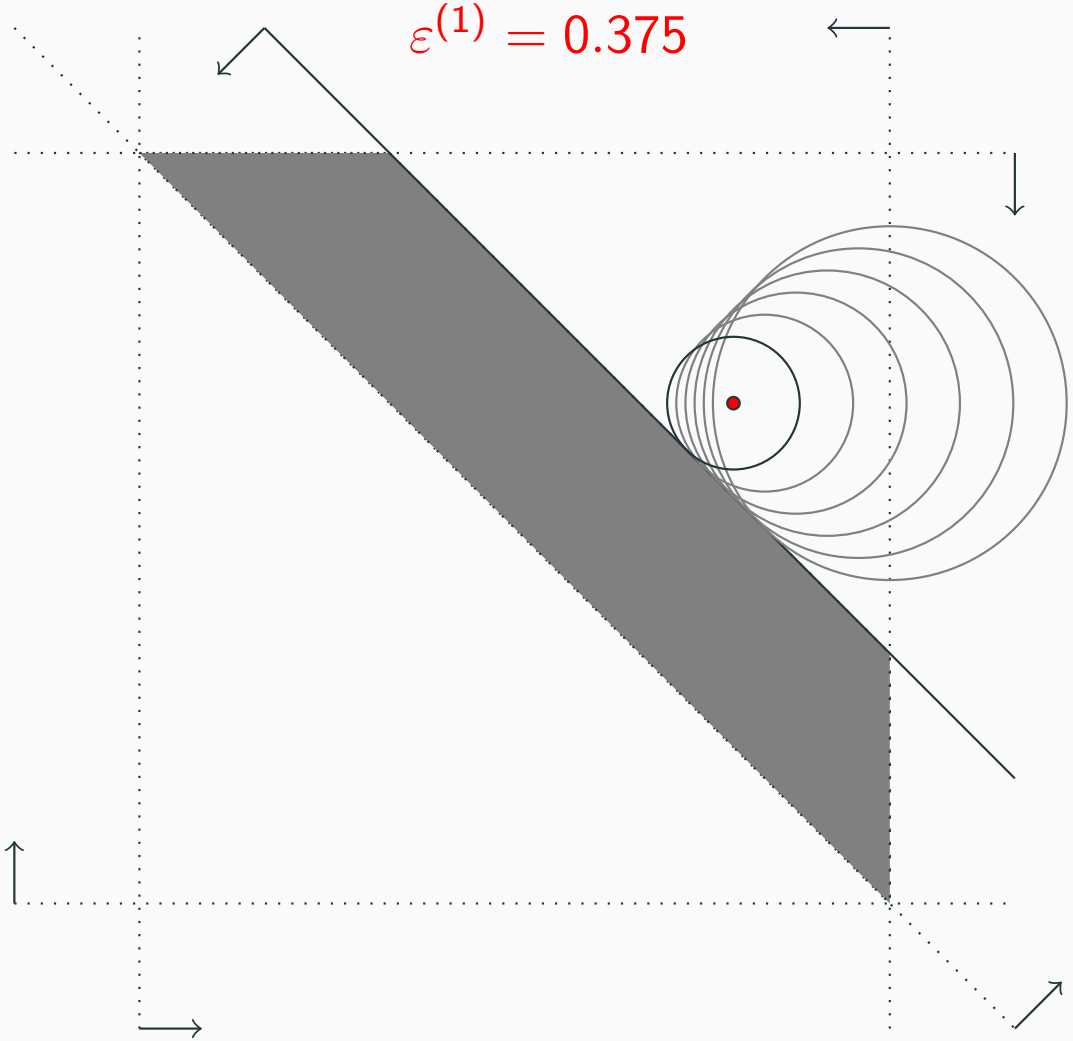


Figure 3: Pivot step

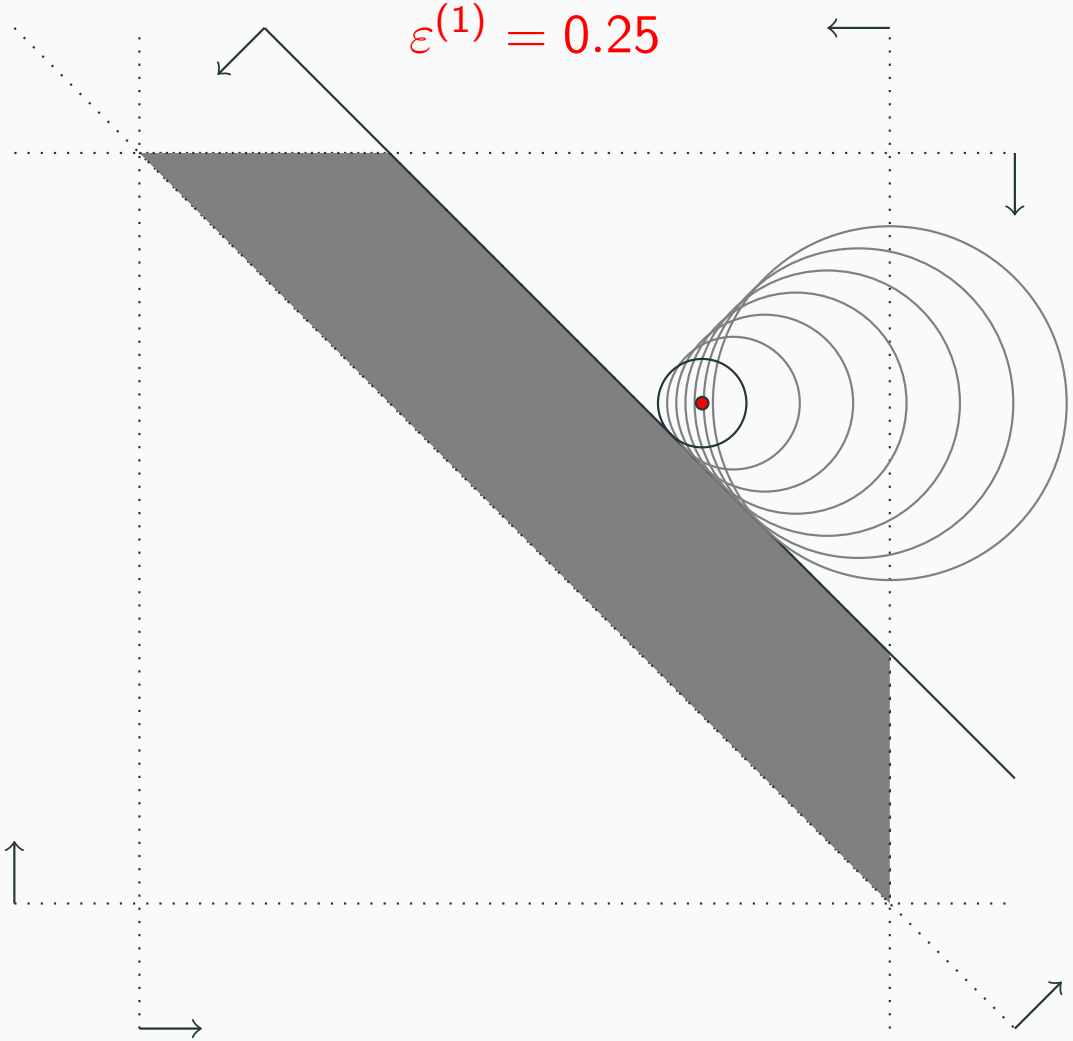


Figure 3: Pivot step

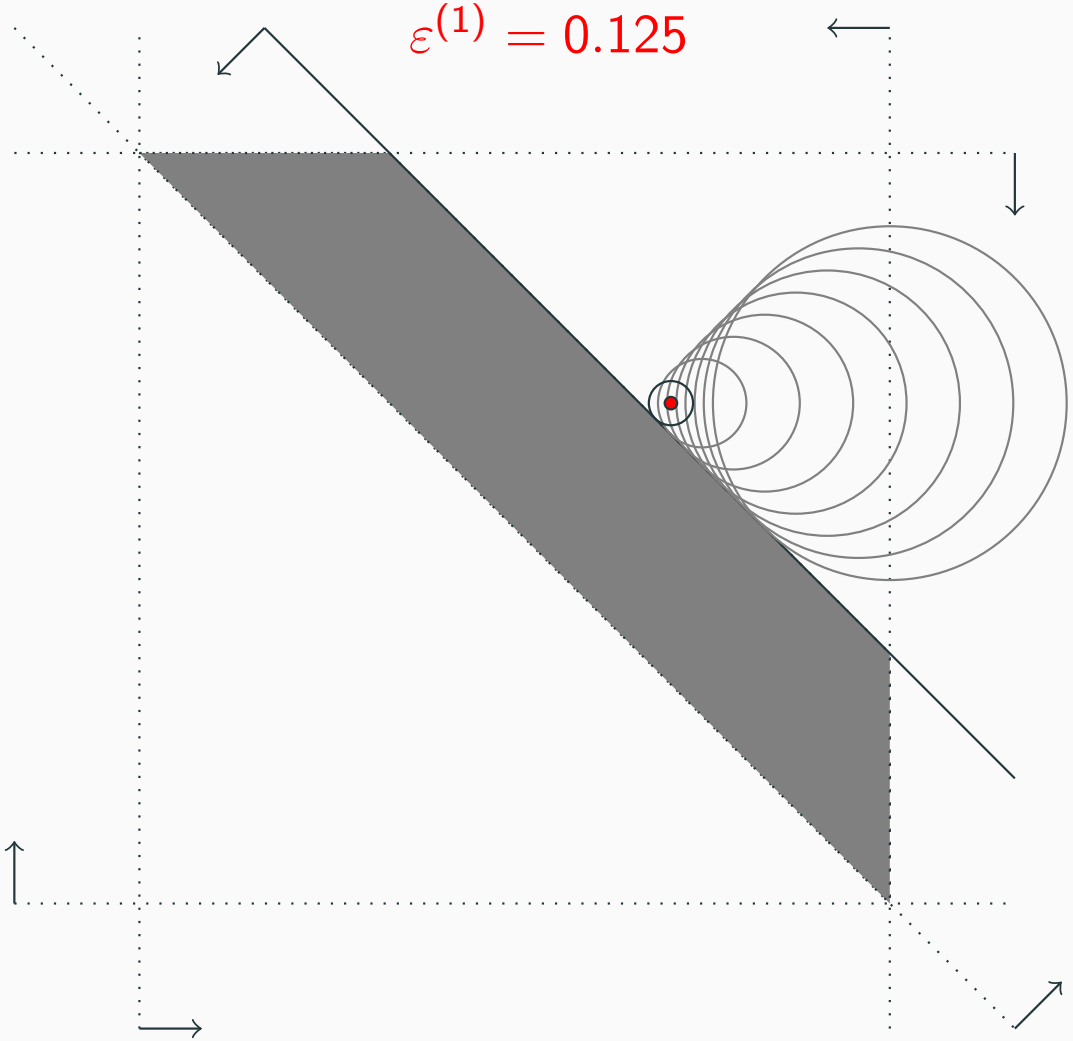


Figure 3: Pivot step

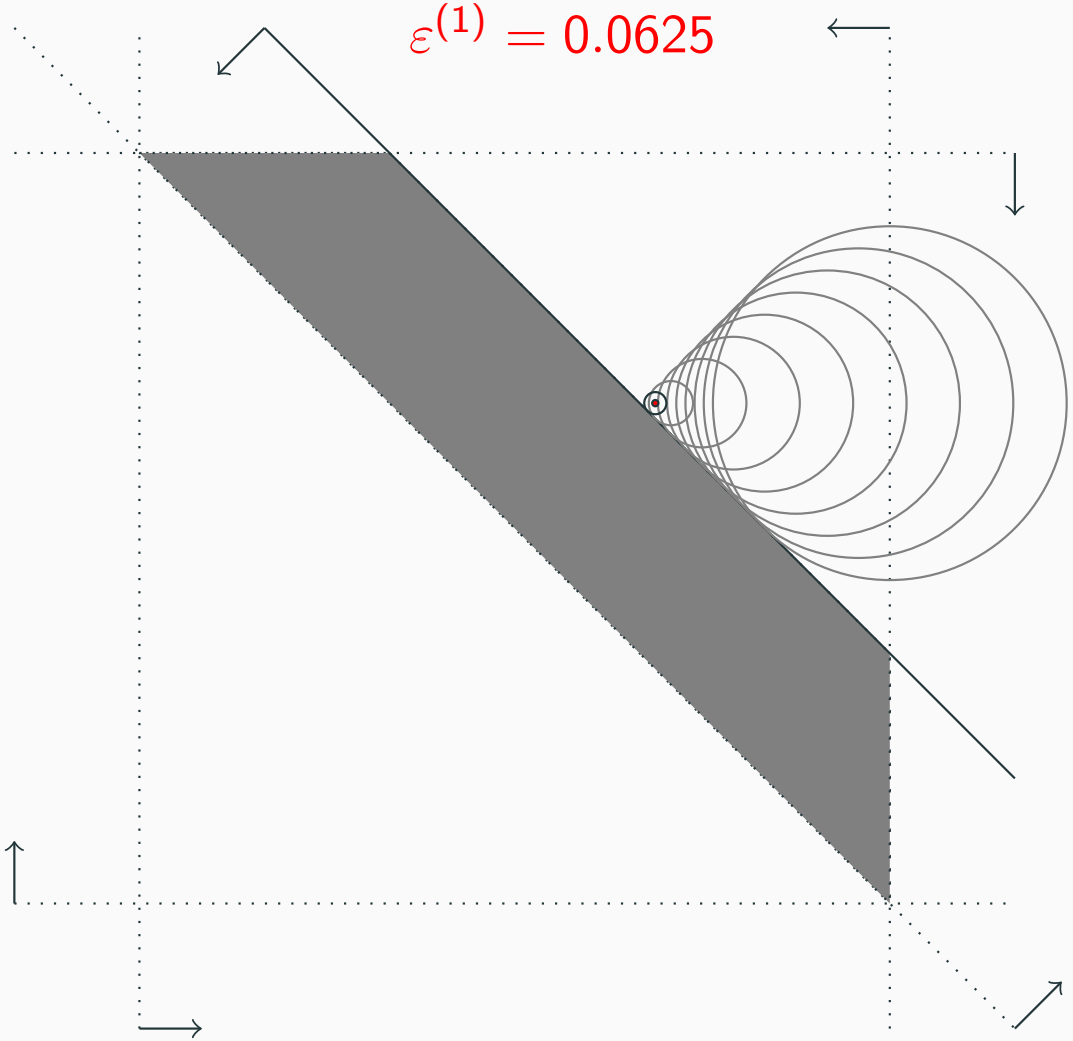


Figure 3: Pivot step

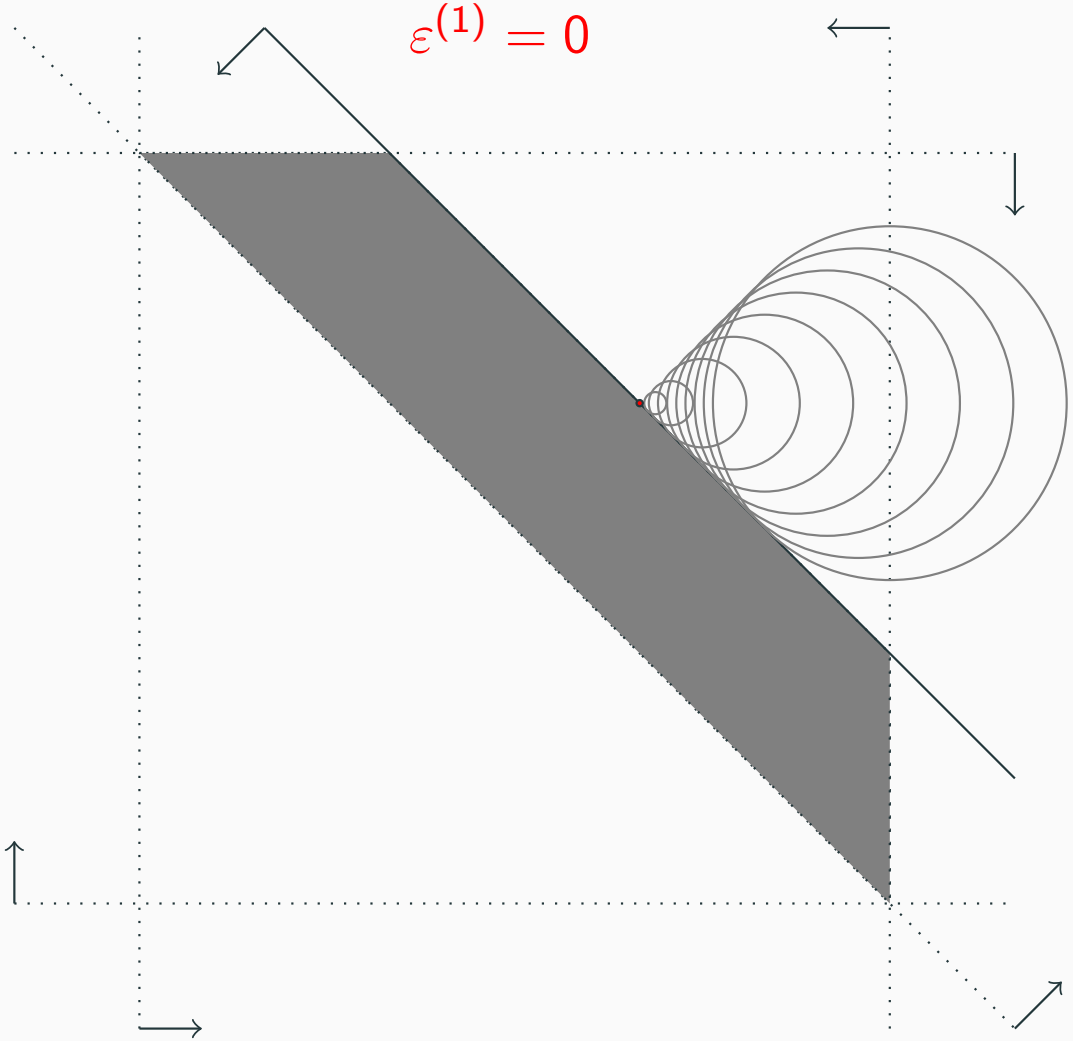


Figure 3: Pivot step

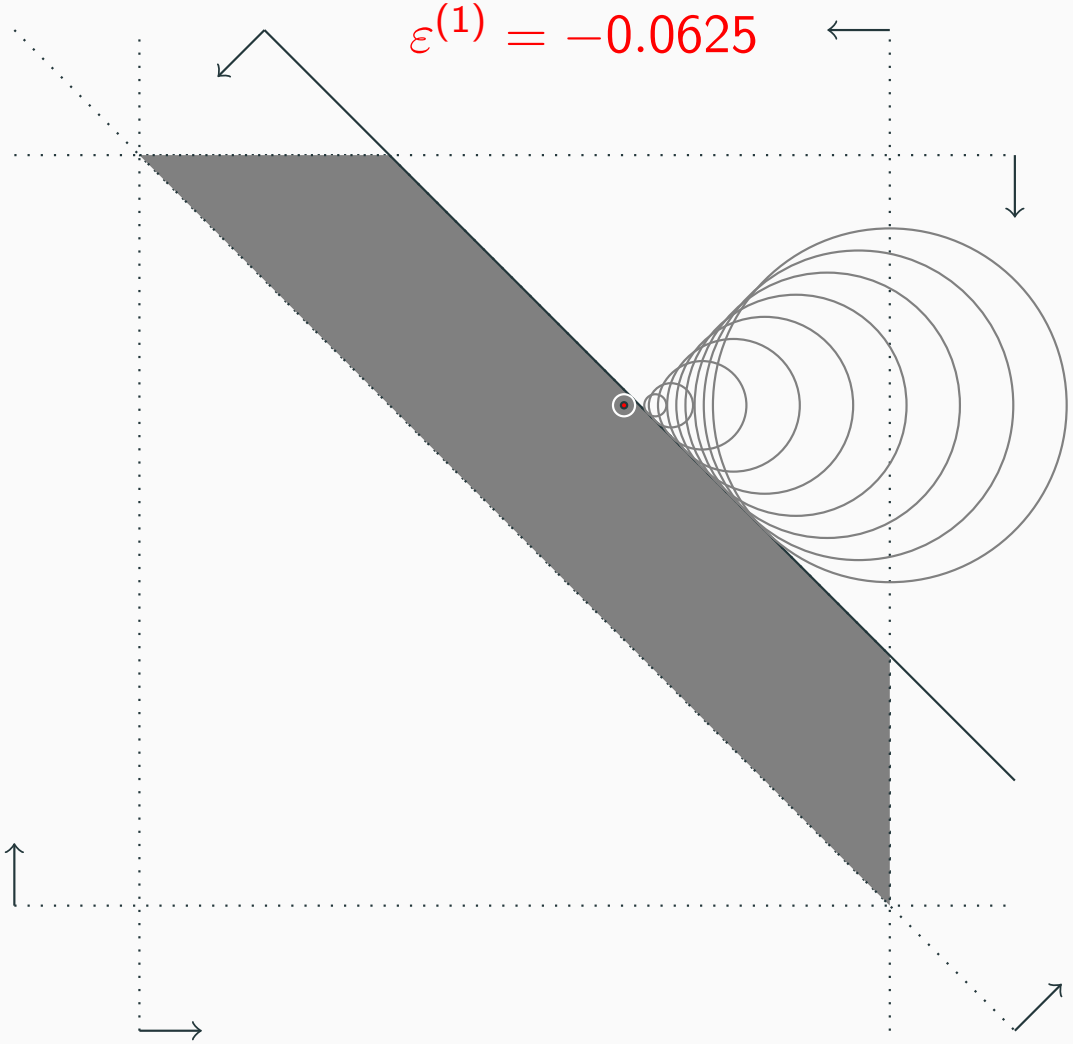


Figure 3: Pivot step

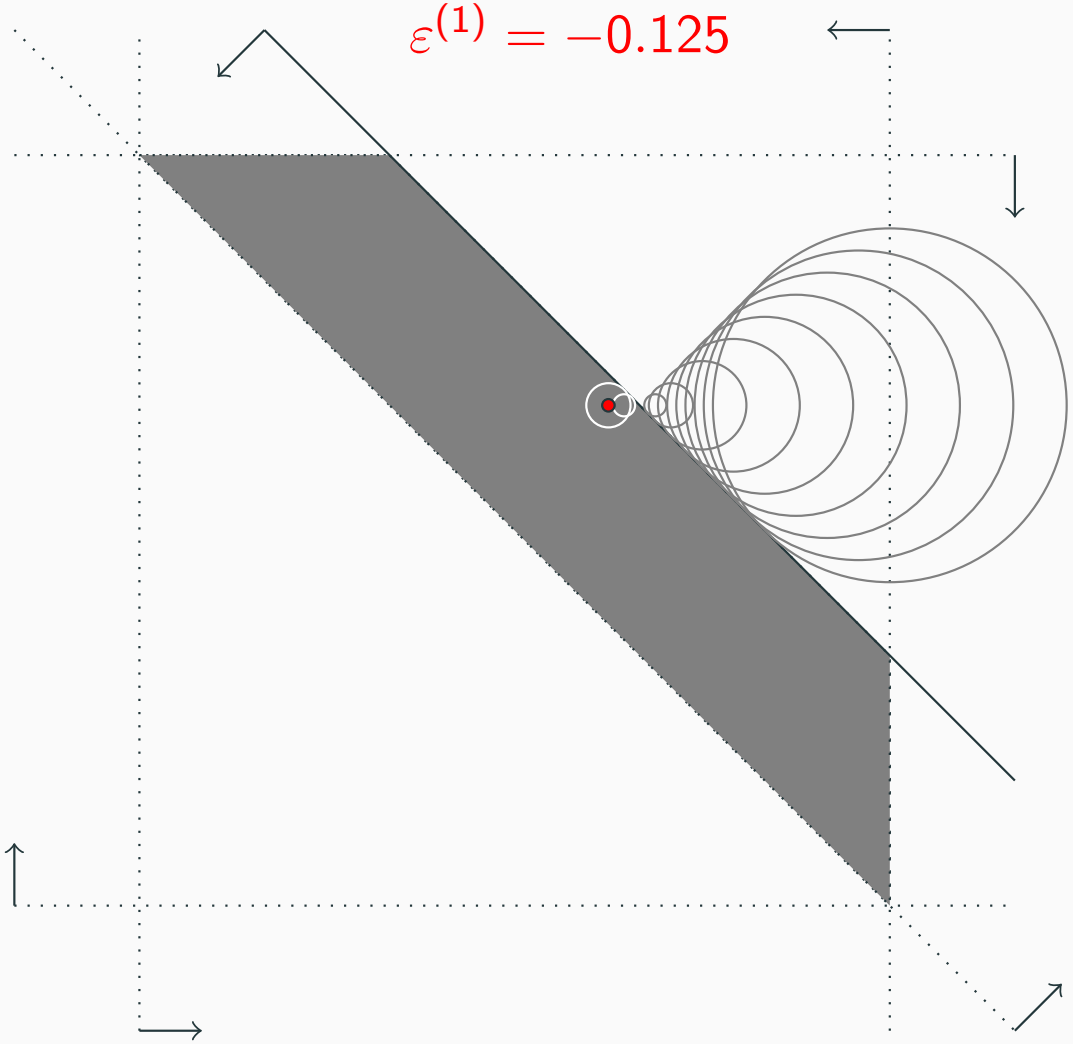


Figure 3: Pivot step

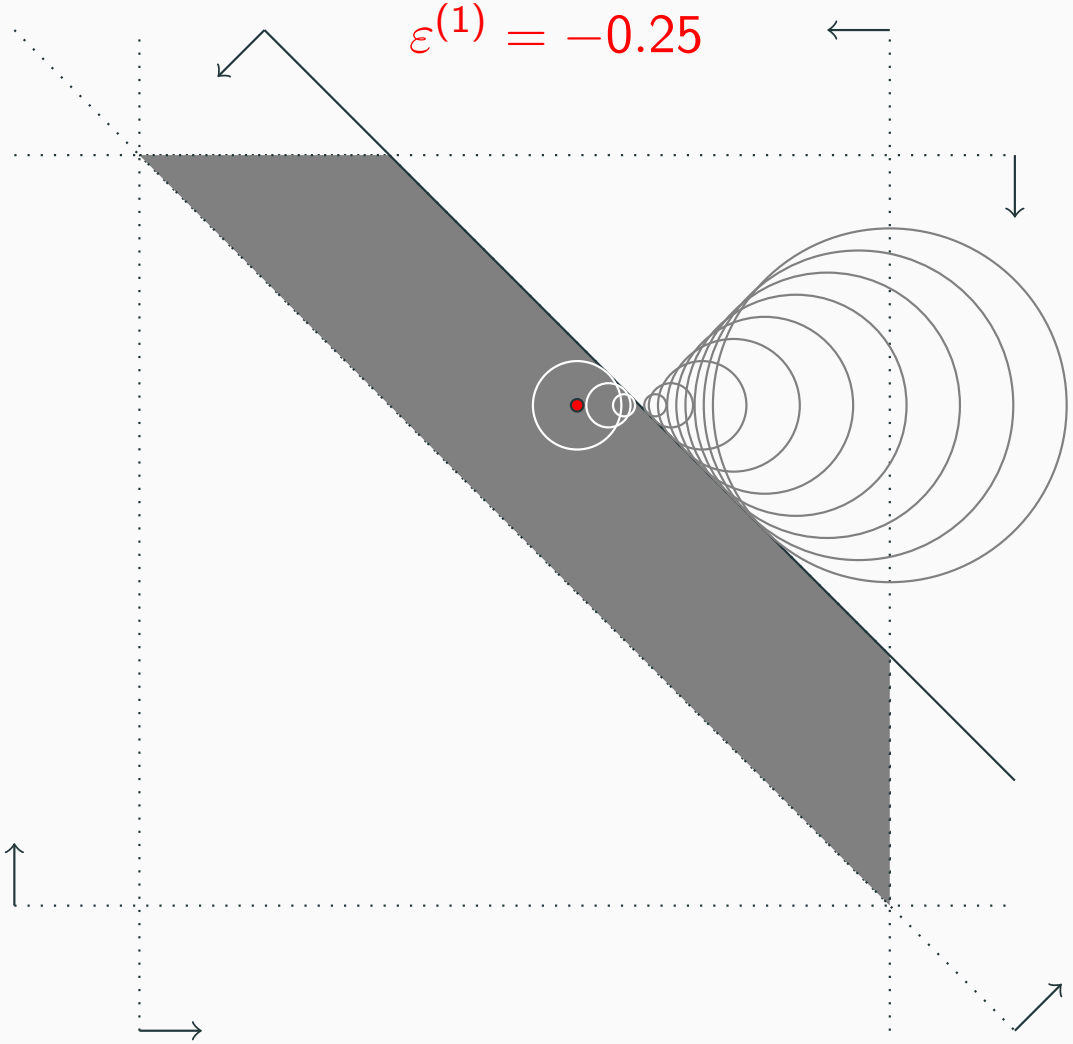


Figure 3: Pivot step

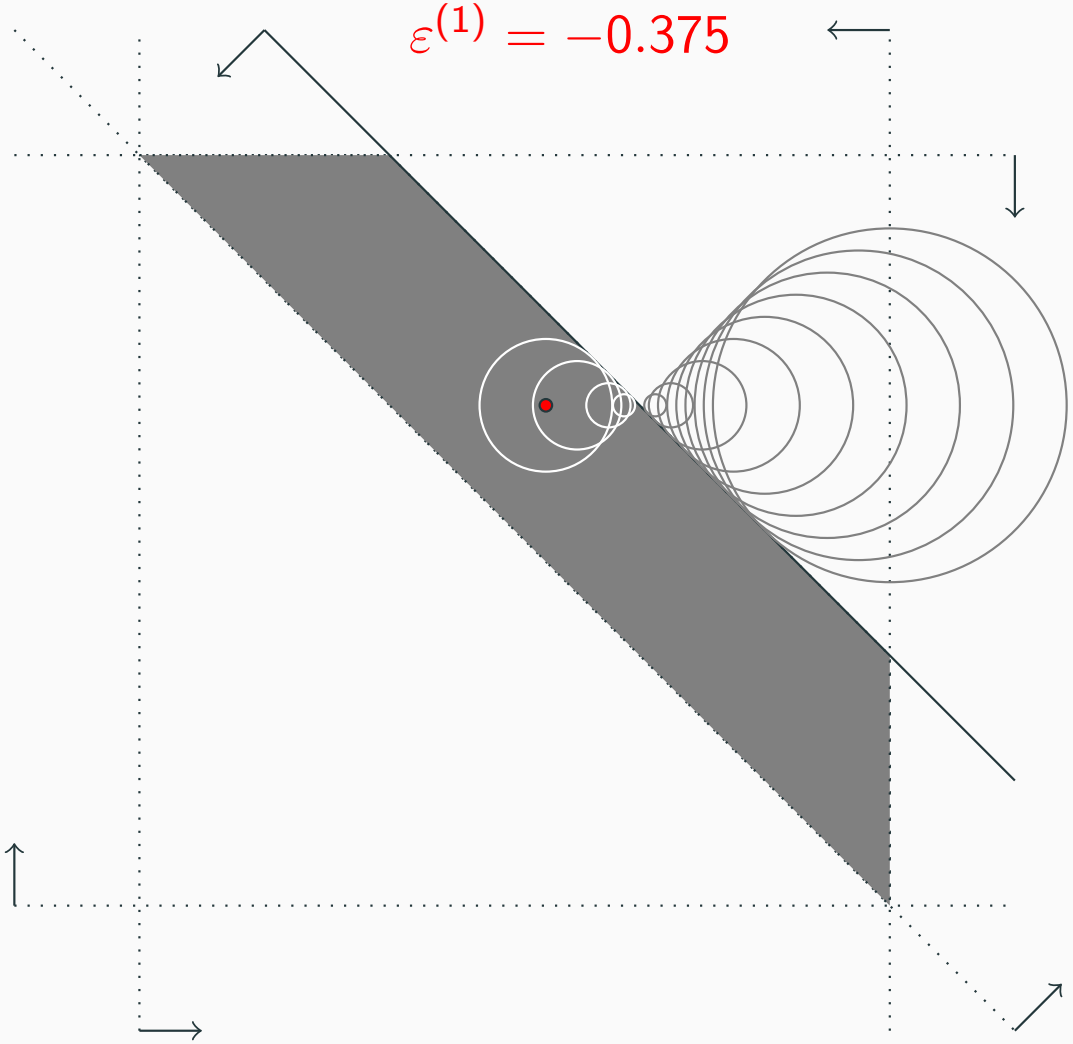


Figure 3: Pivot step

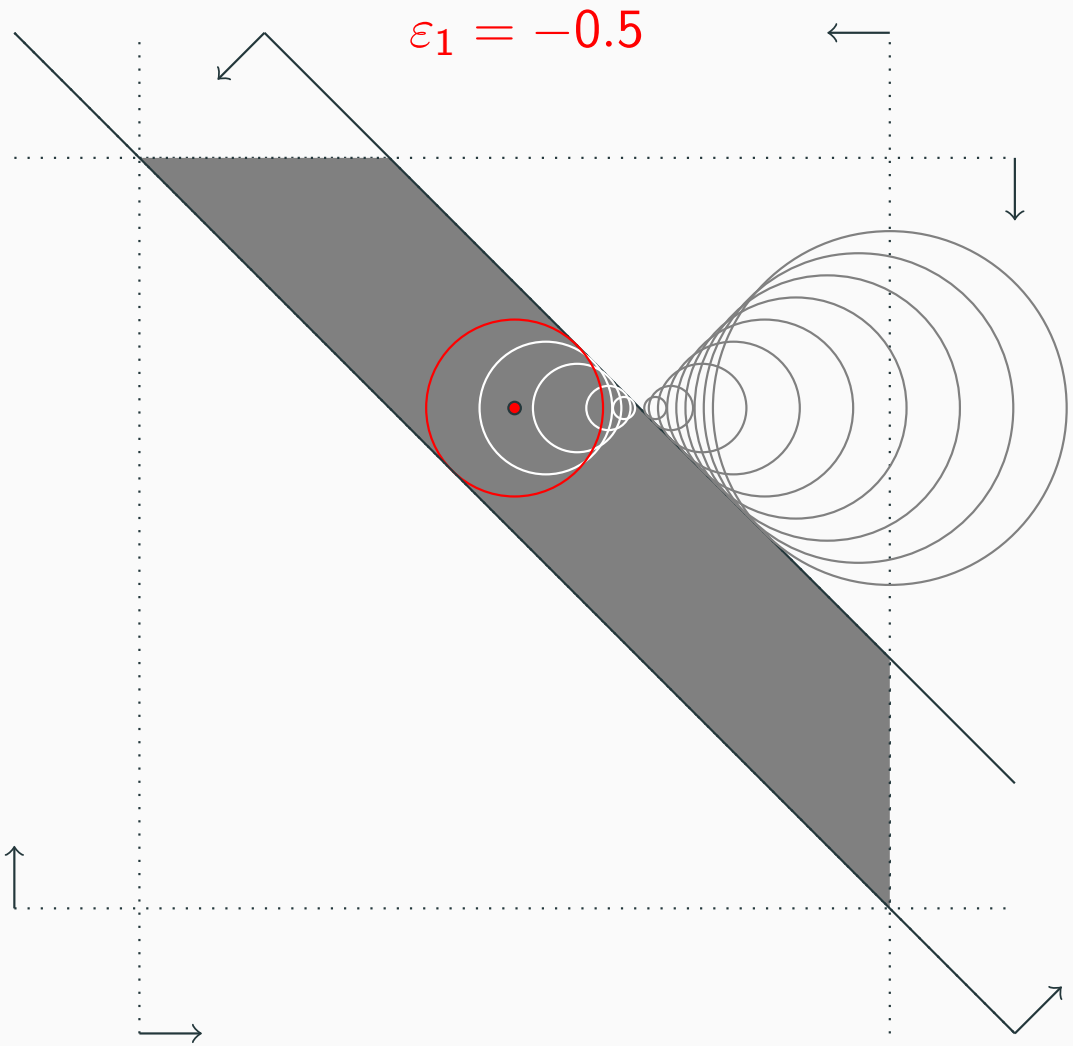


Figure 3: Pivot step

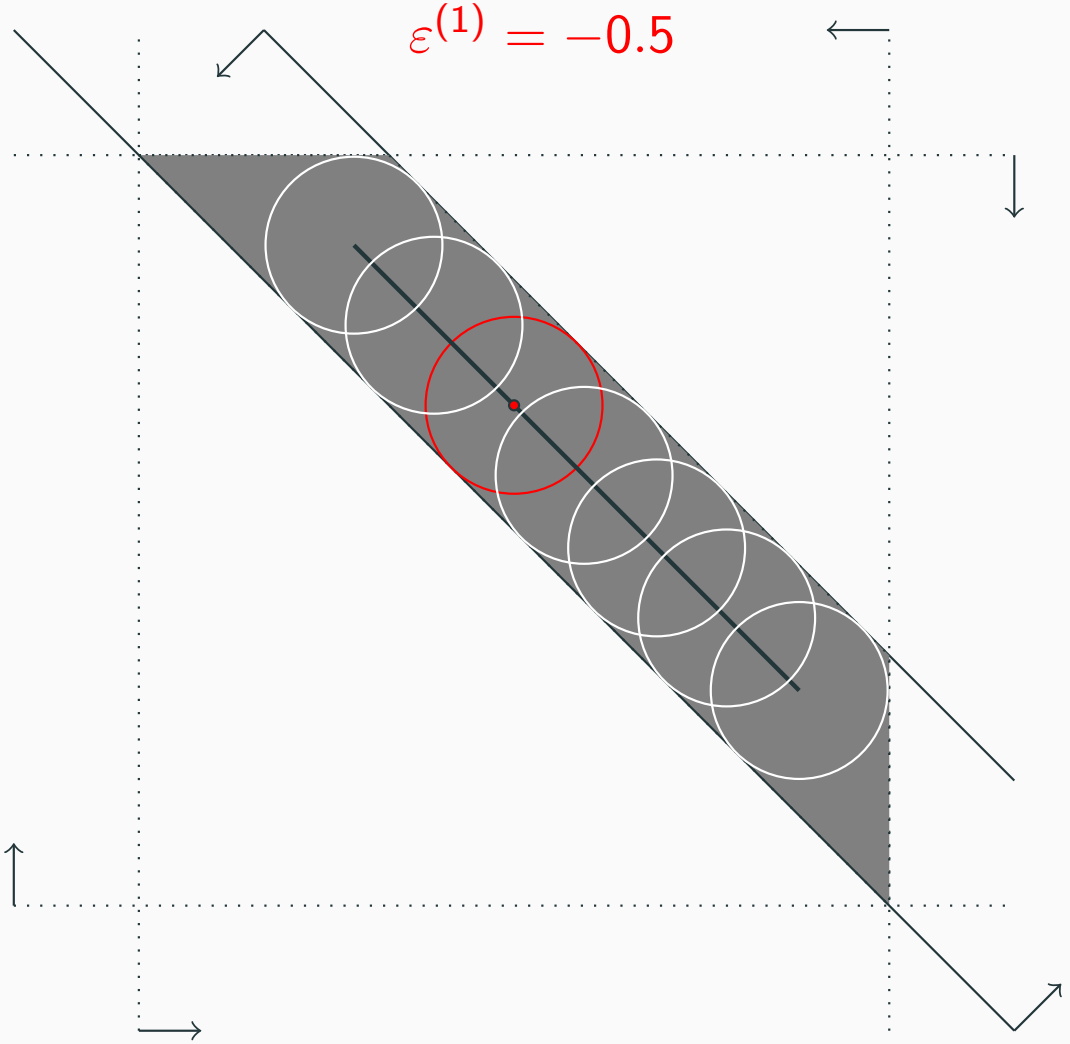


Figure 3: Least core

In the first iteration $d = [\gamma, -1 - \gamma, 1]^\top$.

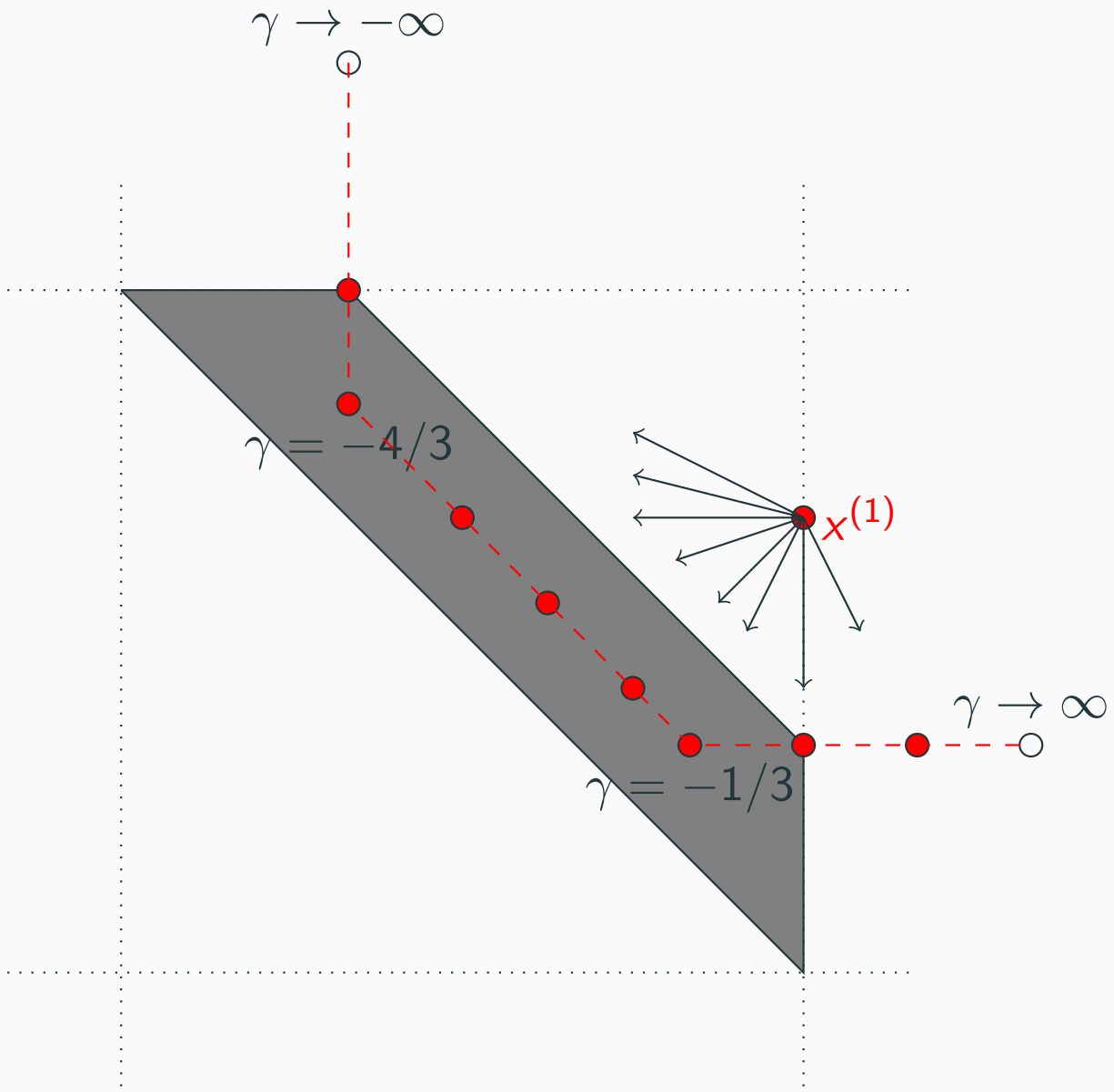


Figure 3: Improving directions

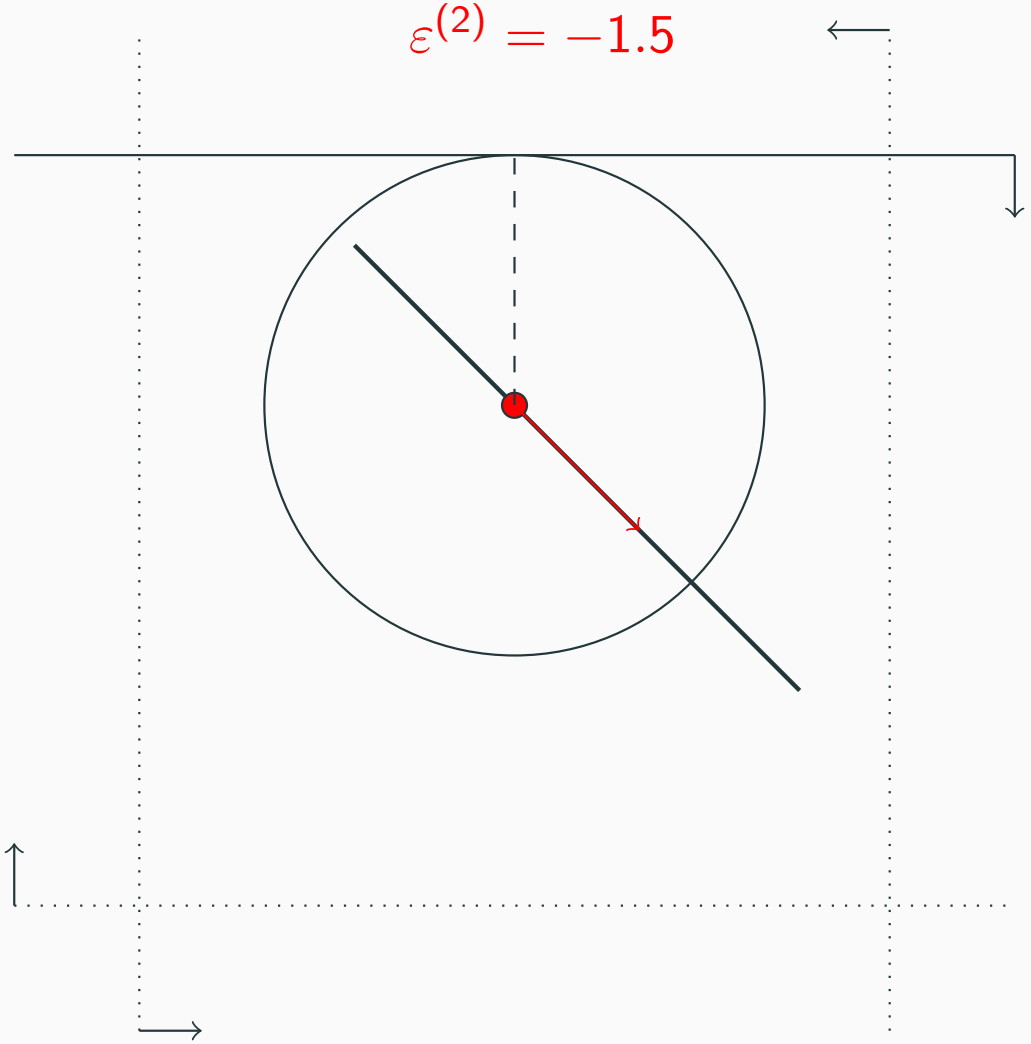


Figure 3: Pivot step

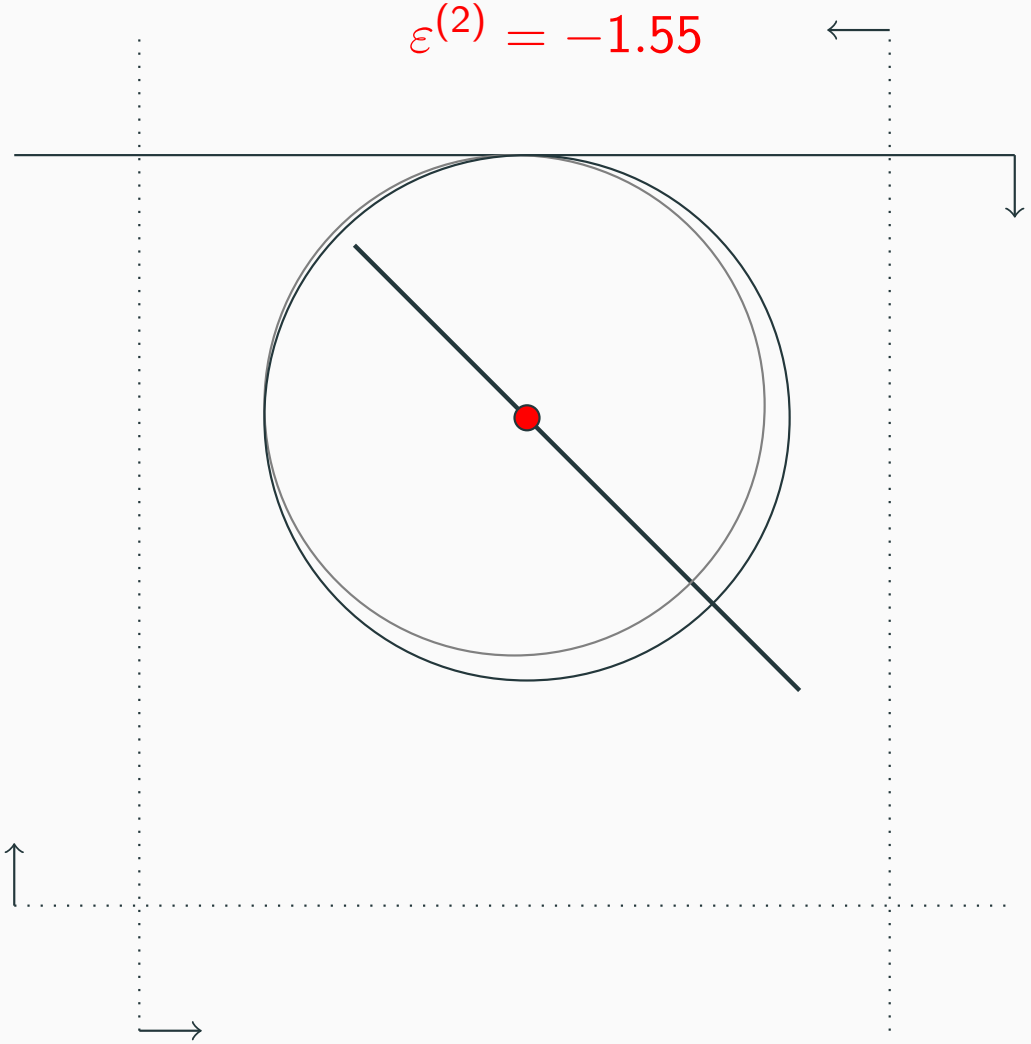


Figure 3: Pivot step

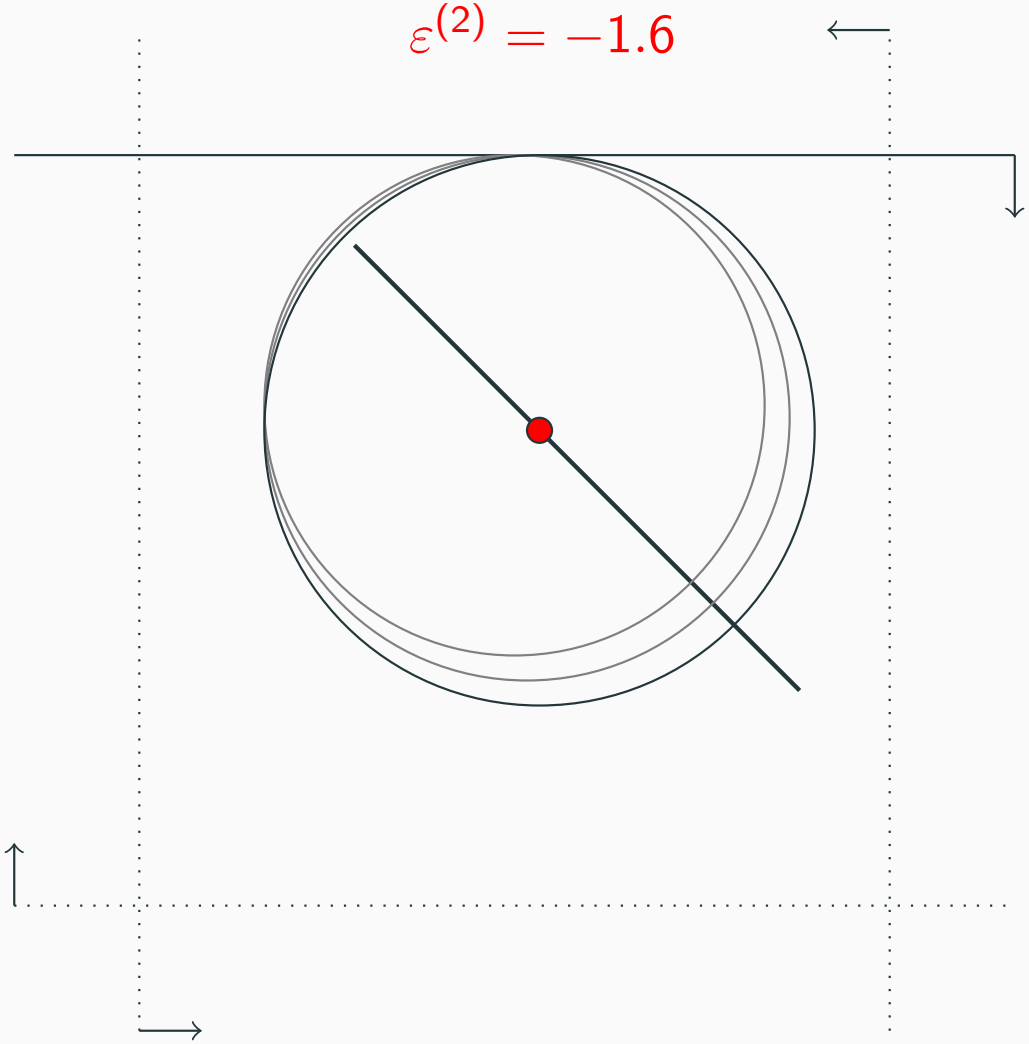


Figure 3: Pivot step

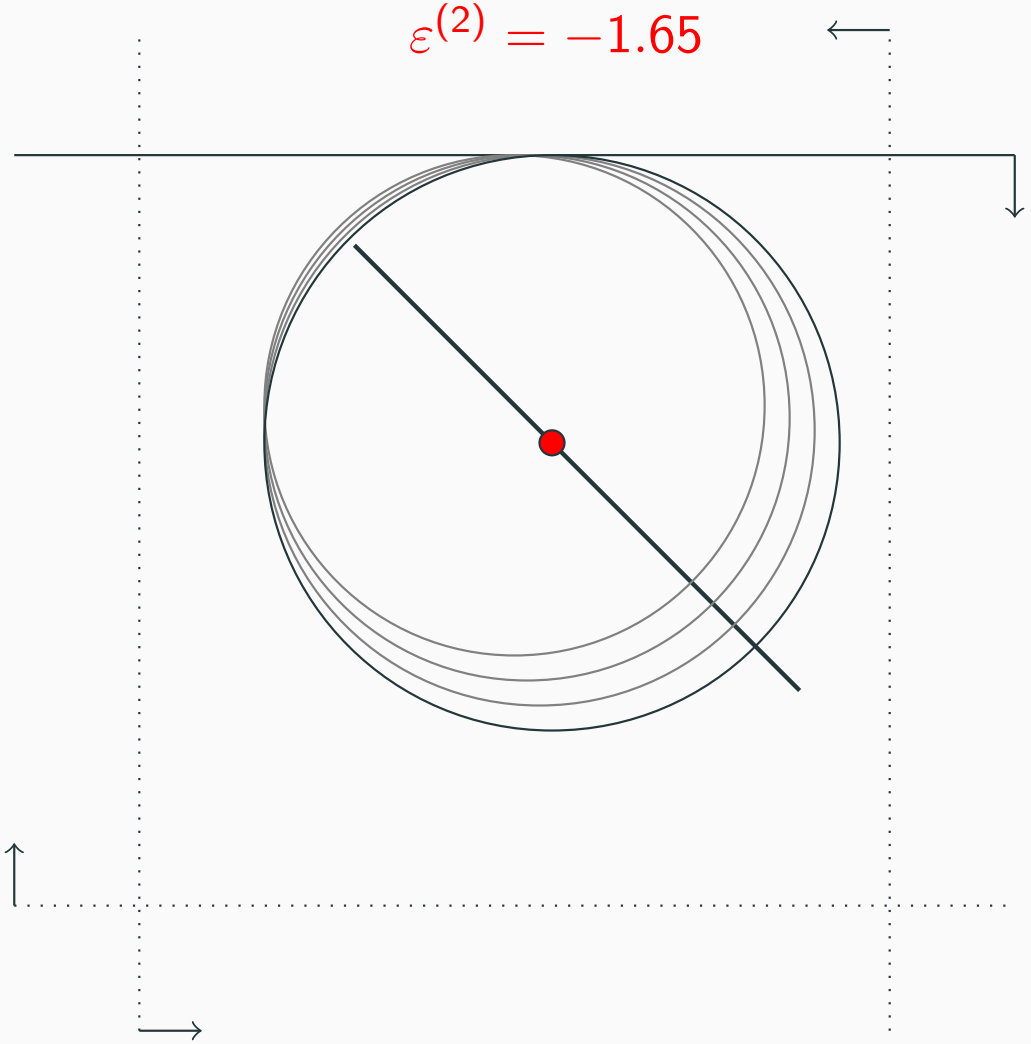


Figure 3: Pivot step

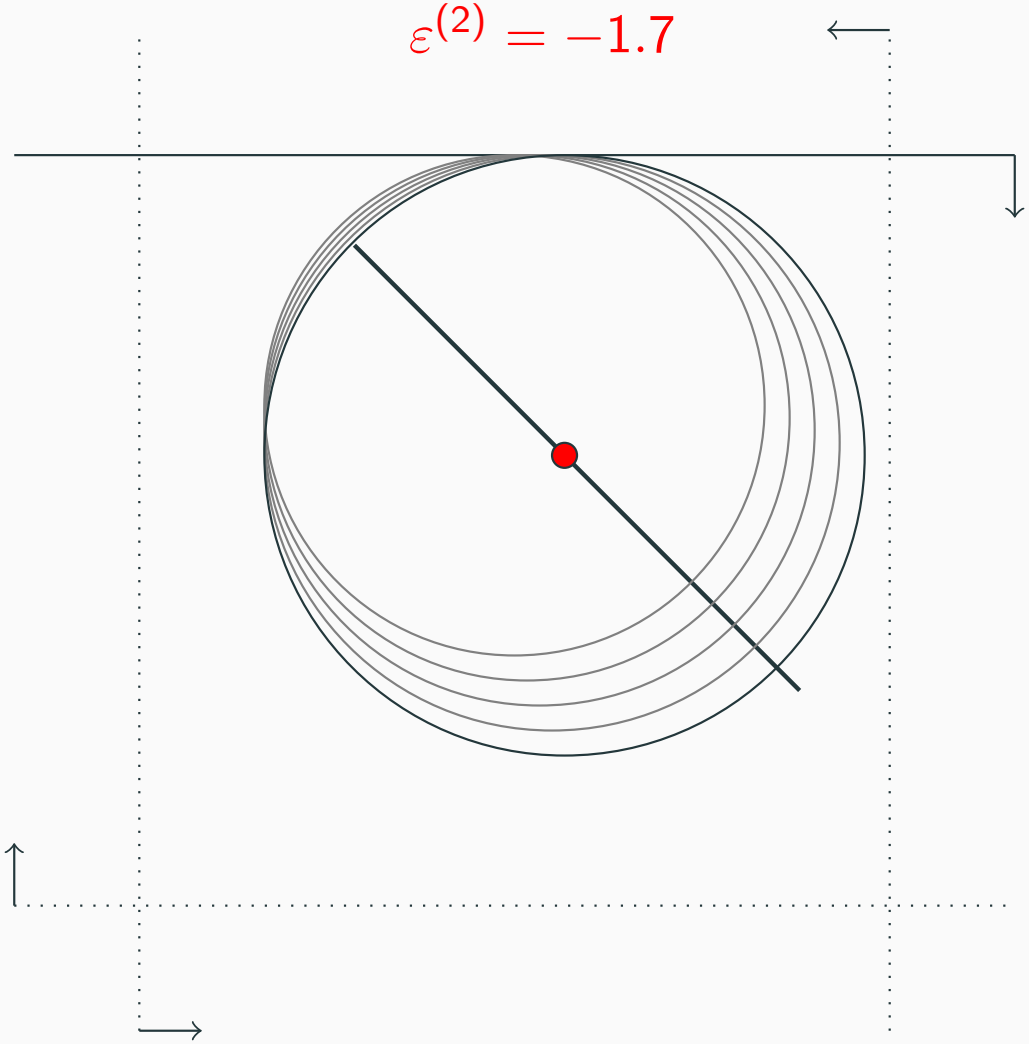


Figure 3: Pivot step

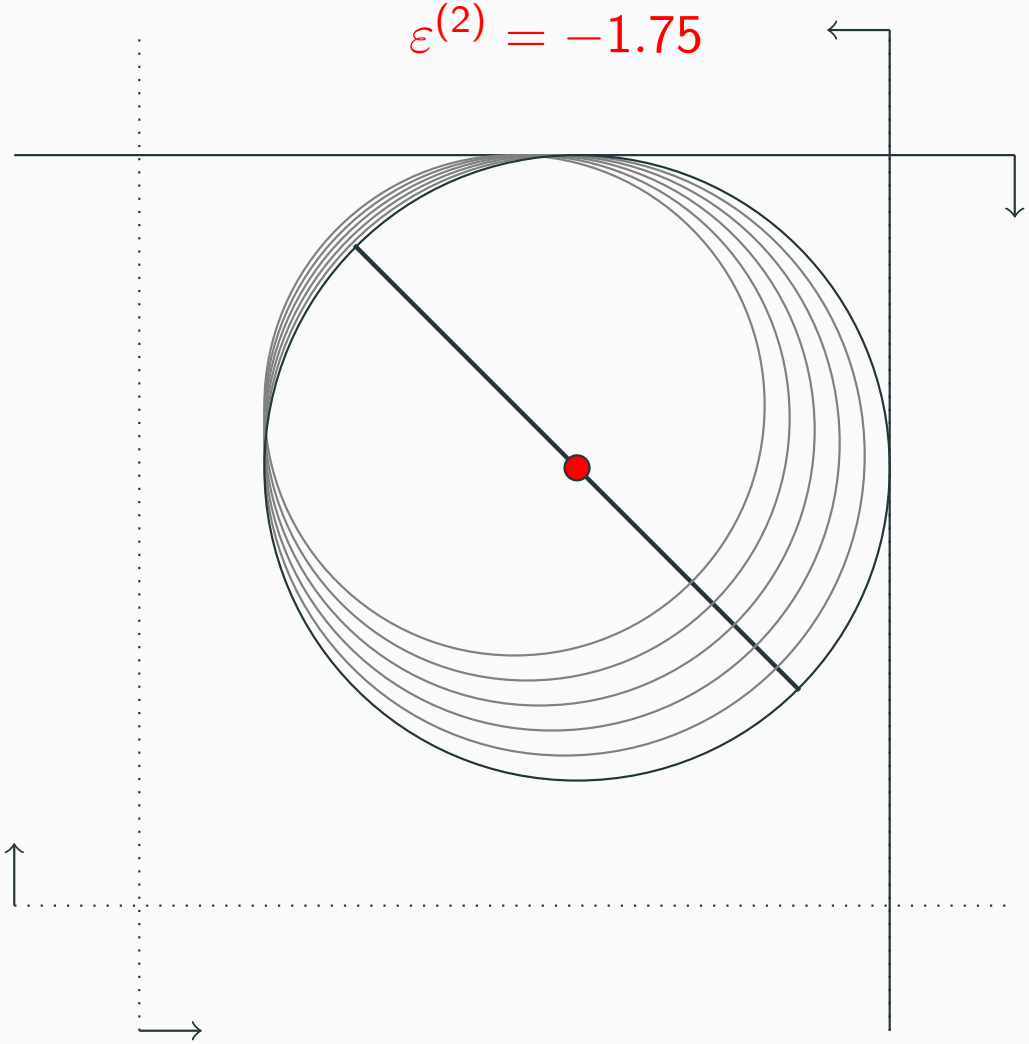


Figure 3: Pivot step

Lexicographical descent

Lemma 2

Let $\mathcal{T} \subsetneq \mathcal{T}_k(x)$ be the largest subcollection with dual feasible support: Algorithm 3 goes through to Step 5. and updates x to $x + \alpha d$. Then we have $\Theta(E(x + \alpha d)) <_L \Theta(E(x))$. Furthermore,

(a) if $\mathcal{T} = \emptyset$, then $\varepsilon_k(x + \alpha d) < \varepsilon_k(x)$,

(b) if $\mathcal{T} \neq \emptyset$, then $\varepsilon_k(x + \alpha d) = \varepsilon_k(x) = \varepsilon_k(\nu)$ and $\mathcal{T}_k(x + \alpha d) = \mathcal{T}_k(\nu)$.

Lexicographical descent

Lemma 2

Let $\mathcal{T} \subsetneq \mathcal{T}_k(x)$ be the largest subcollection with dual feasible support: Algorithm 3 goes through to Step 5. and updates x to $x + \alpha d$. Then we have $\Theta(E(x + \alpha d)) <_L \Theta(E(x))$. Furthermore,

(a) if $\mathcal{T} = \emptyset$, then $\varepsilon_k(x + \alpha d) < \varepsilon_k(x)$,

(b) if $\mathcal{T} \neq \emptyset$, then $\varepsilon_k(x + \alpha d) = \varepsilon_k(x) = \varepsilon_k(\nu)$ and $\mathcal{T}_k(x + \alpha d) = \mathcal{T}_k(\nu)$.

Corollary: no circulation in the payoff space.

Lexicographical descent

Lemma 2

Let $\mathcal{T} \subsetneq \mathcal{T}_k(x)$ be the largest subcollection with dual feasible support: Algorithm 3 goes through to Step 5. and updates x to $x + \alpha d$. Then we have $\Theta(E(x + \alpha d)) <_L \Theta(E(x))$. Furthermore,

(a) if $\mathcal{T} = \emptyset$, then $\varepsilon_k(x + \alpha d) < \varepsilon_k(x)$,

(b) if $\mathcal{T} \neq \emptyset$, then $\varepsilon_k(x + \alpha d) = \varepsilon_k(x) = \varepsilon_k(\nu)$ and $\mathcal{T}_k(x + \alpha d) = \mathcal{T}_k(\nu)$.

Corollary: no circulation in the payoff space.

Main theorem

Algorithm 3 stops after finite steps. The **while** loop executes at most $(n - 1)$ iterations before finding ν

Lexicographical descent

Lemma 2

Let $\mathcal{T} \subsetneq \mathcal{T}_k(x)$ be the largest subcollection with dual feasible support: Algorithm 3 goes through to Step 5. and updates x to $x + \alpha d$. Then we have $\Theta(E(x + \alpha d)) <_L \Theta(E(x))$. Furthermore,

(a) if $\mathcal{T} = \emptyset$, then $\varepsilon_k(x + \alpha d) < \varepsilon_k(x)$,

(b) if $\mathcal{T} \neq \emptyset$, then $\varepsilon_k(x + \alpha d) = \varepsilon_k(x) = \varepsilon_k(\nu)$ and $\mathcal{T}_k(x + \alpha d) = \mathcal{T}_k(\nu)$.

Corollary: no circulation in the payoff space.

Main theorem

Algorithm 3 stops after finite steps. The **while** loop executes at most $(n - 1)$ iterations before finding ν

Reason: no circulation in the tight set space either.

Computational results

Numerical testing

Algorithms

- P, D: primal and dual sequential LP algorithms
- LD: our proposed descent algorithm
  - primal approach $\rightarrow$ minimal iterations
  - strict lexicographical descent
- DK: the only 'other descent algorithm' we know of by [Derks and Kuipers, 1997]
  - dual approach $\rightarrow$ more iterations
  - steps with 0 step size

C++ implementation, tested on Intel Core i5-2500 3.30 GHz CPU with 16 GB RAM, LPs solved by CPLEX 12.7.1's primal simplex

Open-source codes and test instances: [Benedek, 2018] at

`https://github.com/blrzsvrzs/nucleolus`

(version using open-source GLPK solver forthcoming)

Computational results

Table 3: Computing the nucleolus

n	Time				Iterations				Pivots				Subrout.	
	LD	DK	P	D	LD	DK	P	D	LD	DK	P	D	LD	P
5	0.003	0.005	0.002	**0.001**	**1.8**	2.3	**1.8**	2.4	**3.4**	5.5	8	11.7	5.5	**2.9**
10	0.007	0.013	0.007	**0.006**	**1.9**	2.7	**1.9**	2.4	**5.8**	14.5	15.9	42.6	8.5	**3.3**
15	**0.05**	0.2	0.36	0.27	**2**	2.9	**2**	2.5	**8.9**	27.4	71.1	95.7	12.1	**3.6**
20	**2.17**	11.4	20.7	19.1	**2.1**	4.2	**2.1**	3.4	**10.8**	44.4	130	223	15.2	**4.5**
25	**102**	563	OoM	OoT	**2.9**	4.2	OoM	OoT	**16.8**	69.3	OoM	OoT	23.1	OoM
26	**188**	1375	-	-	**3.4**	5.2	-	-	**16.4**	70.5	-	-	22.6	-
27	**486**	3024	-	-	**3.6**	5.4	-	-	**19.7**	78.3	-	-	31	-
28	**3128**	9648	-	-	**3.7**	6	-	-	**19.3**	106	-	-	26.3	-
29	**4665**	22760	-	-	**3.5**	6.1	-	-	**21.3**	89	-	-	32.1	-
30	**8550**	OoT	-	-	**2.8**	OoT	-	-	**18.9**	OoT	-	-	25.4	-

Tackling Problem #3: Linear speed-up

Mostly overlooked: $\forall S \in 2^N \setminus \text{span}(\mathcal{R}^{(k-1)})$

- We argue: replace $\text{span}(\mathcal{R}^{(k-1)})$ with $\cup_{i=0}^{k-1} \mathcal{T}^{(i)*}$
- Theoretically: $\leq n-1 \rightarrow \mathcal{O}(2^n)$
- *Extra* iterations: only if $\mathcal{T}^{(k)*} \subsetneq \text{span}(\mathcal{R}^{(k-1)})$
- *Extra* LPs trivial; expectation:

$$\frac{\text{extra } \#\text{pivots}}{\text{extra } \#\text{iterations}} \approx 0$$

Numerical results

n	Time				Iterations				Pivots			
	LD	DK	P	D	LD	DK	P	D	LD	DK	P	D
10	-4%	12%	2%	-14%	13%	53%	13%	42%	0%	5%	9%	55%
15	-8%	-41%	20%	-3%	22%	99%	22%	83%	0.5%	11%	17%	77%
20	-9%	-51%	27%	27%	31%	155%	31%	99%	2%	6%	19%	102%
21	-2%	-42%	40%	55%	51%	191%	51%	128%	0.4%	19%	27%	141%
22	-7%	-45%	56%	46%	54%	222%	54%	123%	0.5%	19%	22%	122%
23	-1%	-43%	378%	-9%	53%	230%	53%	123%	0%	17%	13%	101%
24	-3%	-42%	-	-	54%	238%	-	-	0.7%	18%	-	-
25	-4%	-59%	-	-	84%	291%	-	-	0.1%	-36%	-	-
26	-0.4%	-51%	-	-	32%	200%	-	-	0%	19%	-	-
27	-7%	-46%	-	-	167%	444%	-	-	0%	27%	-	-
28	-3%	-40%	-	-	51%	328%	-	-	0.5%	26%	-	-
29	3%	-12%	-	-	137%	369%	-	-	0%	35%	-	-
30	3%	-	-	-	64%	-	-	-	0%	-	-	-

Table 4: Change in computing the nucleolus without linear speed-up

	LD	DK	P	D
Ratio	1.3%	5.9%	65%	88%

Numerical results

n	Time				Iterations				Pivots			
	LD	DK	P	D	LD	DK	P	D	LD	DK	P	D
10	-4%	12%	2%	-14%	13%	53%	13%	42%	0%	5%	9%	55%
15	-8%	-41%	20%	-3%	22%	99%	22%	83%	0.5%	11%	17%	77%
20	-9%	-51%	27%	27%	31%	155%	31%	99%	2%	6%	19%	102%
21	-2%	-42%	40%	55%	51%	191%	51%	128%	0.4%	19%	27%	141%
22	-7%	-45%	56%	46%	54%	222%	54%	123%	0.5%	19%	22%	122%
23	-1%	-43%	378%	-9%	53%	230%	53%	123%	0%	17%	13%	101%
24	-3%	-42%	-	-	54%	238%	-	-	0.7%	18%	-	-
25	-4%	-59%	-	-	84%	291%	-	-	0.1%	-36%	-	-
26	-0.4%	-51%	-	-	32%	200%	-	-	0%	19%	-	-
27	-7%	-46%	-	-	167%	444%	-	-	0%	27%	-	-
28	-3%	-40%	-	-	51%	328%	-	-	0.5%	26%	-	-
29	3%	-12%	-	-	137%	369%	-	-	0%	35%	-	-
30	3%	-	-	-	64%	-	-	-	0%	-	-	-

Table 4: Change in computing the nucleolus without linear speed-up

	LD	DK	P	D	
Ratio	1.3%	5.9%	65%	88%	Trade-off: warm start!

Numerical results

n	Time				Iterations				Pivots			
	LD	DK	P	D	LD	DK	P	D	LD	DK	P	D
10	-4%	12%	2%	-14%	13%	53%	13%	42%	0%	5%	9%	55%
15	-8%	-41%	20%	-3%	22%	99%	22%	83%	0.5%	11%	17%	77%
20	-9%	-51%	27%	27%	31%	155%	31%	99%	2%	6%	19%	102%
21	-2%	-42%	40%	55%	51%	191%	51%	128%	0.4%	19%	27%	141%
22	-7%	-45%	56%	46%	54%	222%	54%	123%	0.5%	19%	22%	122%
23	-1%	-43%	378%	-9%	53%	230%	53%	123%	0%	17%	13%	101%
24	-3%	-42%	-	-	54%	238%	-	-	0.7%	18%	-	-
25	-4%	-59%	-	-	84%	291%	-	-	0.1%	-36%	-	-
26	-0.4%	-51%	-	-	32%	200%	-	-	0%	19%	-	-
27	-7%	-46%	-	-	167%	444%	-	-	0%	27%	-	-
28	-3%	-40%	-	-	51%	328%	-	-	0.5%	26%	-	-
29	3%	-12%	-	-	137%	369%	-	-	0%	35%	-	-
30	3%	-	-	-	64%	-	-	-	0%	-	-	-

Table 4: Change in computing the nucleolus without linear speed-up

	LD	DK	P	D	
Ratio	1.3%	5.9%	65%	88%	Trade-off: warm start!
Time	-4.3%	-38%	87%	17%	Time decrease!

Summary

Takeaways

- Cooperative game theory introduction
- Lexicographical optimisation with LPs
- Trade-offs: different formulations
- Hard problems: relaxation can be just as good
- Efficient algorithm computing the nucleolus
    - both in CPU time and memory

Thank you for the attention!

Questions?

(now or later: benedek.marton@krtk.mta.hu)

References i

Benedek, M. (2018).
Nucleolus.
GitHub repository.

Benedek, M., Fliege, J., and Nguyen, T.-D. (2020).
Finding and verifying the nucleolus of cooperative games.
Mathematical Programming.

Derks, J. and Kuipers, J. (1997).
Implementing the simplex method for computing the prenucleolus of transferable utility games.

Dragan, I. (1981).
A procedure for finding the nucleolus of a cooperativen person game.
Zeitschrift für Operations Research, 25(5):119–131.

References ii


Freund, R. M., Roundy, R., and Todd, M. J. (1985).
Identifying the set of always-active constraints in a system of linear inequalities by a single linear program robert m. freund.
Sloan W.P., (1674-85).


Kido, K. (2008).
A modified kohlberg criterion and a nonlinear method to compute the nucleolus of a cooperative game.
Taiwanese Journal of Mathematics, pages 1581–1590.


Kohlberg, E. (1971).
On the nucleolus of a characteristic function game.
SIAM Journal on Applied Mathematics, 1(20):62–66.


Kohlberg, E. (1972).
The nucleolus as a solution of a minimization problem.
SIAM Journal on Applied Mathematics, 23(1):34–39.

References iii


Maschler, M., Peleg, B., and Shapley, L. S. (1979).
Geometric properties of the kernel, nucleolus, and related solution concepts.
Mathematics of operations research, 4(4):303–338.


Nguyen, T.-D. and Thomas, L. (2016).
Finding the nucleoli of large cooperative games.
European Journal of Operational Research, 3(248):1078–1092.


Owen, G. (1974).
A note on the nucleolus.
International Journal of Game Theory, 3(2):101–103.


Potters, J. A., Reijnierse, J. H., and Ansing, M. (1996).
Computing the nucleolus by solving a prolonged simplex algorithm.
Mathematics of operations research, 21(3):757–768.

References iv


Puerto, J. and Perea, F. (2013).
Finding the nucleolus of any n-person cooperative game by a single linear program.
Computers & Operations Research, 40(10):2308–2313.


Reijnierse, J. H. (1995).
Games, graphs and algorithms.
Ph.D. Thesis, Nijmegen.


Sankaran, J. K. (1991).
On finding the nucleolus of an n-person cooperative game.
International Journal of Game Theory, 19(4):329–338.


Schmeidler, D. (1969).
The nucleolus of a characteristic function game.
SIAM Journal of Applied Mathematics, 17(6):1163–1170.

References v


Shapley, L. S. (1953).
A value for *n*-person games. In: Kuhn HW and Tucker AW (eds.) Contributions to the Theory of Games II, volume 28 of Annals of Mathematics Studies, pages 307–317.
Princeton University Press, Princeton.


Shapley, L. S. (1955).
Markets as cooperative games.
Technical report, RAND CORP SANTA MONICA CA.


Solymosi, T. (1993).
On computing the nucleolus of cooperative games.

Appendix

Lexicographical descent

Lemma 2

Let $\mathcal{T} \subsetneq \mathcal{T}_k(x)$ be the largest subcollection with dual feasible support: Algorithm 3 goes through to Step 5. and updates x to $x + \alpha d$. Then we have $\Theta(E(x + \alpha d)) <_L \Theta(E(x))$. Furthermore,

(a) if $\mathcal{T} = \emptyset$, then $\varepsilon_k(x + \alpha d) < \varepsilon_k(x)$,

(b) if $\mathcal{T} \neq \emptyset$, then $\varepsilon_k(x + \alpha d) = \varepsilon_k(x) = \varepsilon_k(\nu)$ and $\mathcal{T}_k(x + \alpha d) = \mathcal{T}_k(\nu)$.

Lexicographical descent

Lemma 2

Let $\mathcal{T} \subsetneq \mathcal{T}_k(x)$ be the largest subcollection with dual feasible support: Algorithm 3 goes through to Step 5. and updates x to $x + \alpha d$. Then we have $\Theta(E(x + \alpha d)) <_L \Theta(E(x))$. Furthermore,

(a) if $\mathcal{T} = \emptyset$, then $\varepsilon_k(x + \alpha d) < \varepsilon_k(x)$,

(b) if $\mathcal{T} \neq \emptyset$, then $\varepsilon_k(x + \alpha d) = \varepsilon_k(x) = \varepsilon_k(\nu)$ and $\mathcal{T}_k(x + \alpha d) = \mathcal{T}_k(\nu)$.

Proof: $x + \alpha d \in pi(v)$ if $x \in pi(v)$

Lexicographical descent

Lemma 2

Let $\mathcal{T} \subsetneq \mathcal{T}_k(x)$ be the largest subcollection with dual feasible support: Algorithm 3 goes through to Step 5. and updates x to $x + \alpha d$. Then we have $\Theta(E(x + \alpha d)) <_L \Theta(E(x))$. Furthermore,

(a) if $\mathcal{T} = \emptyset$, then $\varepsilon_k(x + \alpha d) < \varepsilon_k(x)$,

(b) if $\mathcal{T} \neq \emptyset$, then $\varepsilon_k(x + \alpha d) = \varepsilon_k(x) = \varepsilon_k(\nu)$ and $\mathcal{T}_k(x + \alpha d) = \mathcal{T}_k(\nu)$.

Proof: $x + \alpha d \in pi(v)$ if $x \in pi(v)$

- $x(S) = (x + \alpha d)(S)$ for all $S \in \text{span}(\mathcal{R}_{k-1})$

Lexicographical descent

Lemma 2

Let $\mathcal{T} \subsetneq \mathcal{T}_k(x)$ be the largest subcollection with dual feasible support: Algorithm 3 goes through to Step 5. and updates x to $x + \alpha d$. Then we have $\Theta(E(x + \alpha d)) <_L \Theta(E(x))$. Furthermore,

(a) if $\mathcal{T} = \emptyset$, then $\varepsilon_k(x + \alpha d) < \varepsilon_k(x)$,

(b) if $\mathcal{T} \neq \emptyset$, then $\varepsilon_k(x + \alpha d) = \varepsilon_k(x) = \varepsilon_k(\nu)$ and $\mathcal{T}_k(x + \alpha d) = \mathcal{T}_k(\nu)$.

Proof: $x + \alpha d \in pi(v)$ if $x \in pi(v)$

- $x(S) = (x + \alpha d)(S)$ for all $S \in \text{span}(\mathcal{R}_{k-1})$
- $\Theta(E(x))$ and $\Theta(E(x + \alpha d))$ both starts with the same values $E(S, \nu) \geq \varepsilon_{k-1}(\nu)$

Lexicographical descent

Lemma 2

Let $\mathcal{T} \subsetneq \mathcal{T}_k(x)$ be the largest subcollection with dual feasible support: Algorithm 3 goes through to Step 5. and updates x to $x + \alpha d$. Then we have $\Theta(E(x + \alpha d)) <_L \Theta(E(x))$. Furthermore,

(a) if $\mathcal{T} = \emptyset$, then $\varepsilon_k(x + \alpha d) < \varepsilon_k(x)$,

(b) if $\mathcal{T} \neq \emptyset$, then $\varepsilon_k(x + \alpha d) = \varepsilon_k(x) = \varepsilon_k(\nu)$ and $\mathcal{T}_k(x + \alpha d) = \mathcal{T}_k(\nu)$.

Proof: $x + \alpha d \in pi(v)$ if $x \in pi(v)$

- $x(S) = (x + \alpha d)(S)$ for all $S \in \text{span}(\mathcal{R}_{k-1})$
- $\Theta(E(x))$ and $\Theta(E(x + \alpha d))$ both starts with the same values $E(S, \nu) \geq \varepsilon_{k-1}(\nu)$
- focus on truncated excess vectors: $2^N \setminus \text{span}(\mathcal{R}_{k-1})$

Lexicographical descent

Lemma 2

Let $\mathcal{T} \subsetneq \mathcal{T}_k(x)$ be the largest subcollection with dual feasible support: Algorithm 3 goes through to Step 5. and updates x to $x + \alpha d$. Then we have $\Theta(E(x + \alpha d)) <_L \Theta(E(x))$. Furthermore,

(a) if $\mathcal{T} = \emptyset$, then $\varepsilon_k(x + \alpha d) < \varepsilon_k(x)$,

(b) if $\mathcal{T} \neq \emptyset$, then $\varepsilon_k(x + \alpha d) = \varepsilon_k(x) = \varepsilon_k(\nu)$ and $\mathcal{T}_k(x + \alpha d) = \mathcal{T}_k(\nu)$.

Proof: $x + \alpha d \in pi(v)$ if $x \in pi(v)$

- $x(S) = (x + \alpha d)(S)$ for all $S \in \text{span}(\mathcal{R}_{k-1})$
- $\Theta(E(x))$ and $\Theta(E(x + \alpha d))$ both starts with the same values $E(S, \nu) \geq \varepsilon_{k-1}(\nu)$
- focus on truncated excess vectors: $2^N \setminus \text{span}(\mathcal{R}_{k-1})$

(a): next value in both corresponds to some $S \in \mathcal{T}_k$

Lexicographical descent

Lemma 2

Let $\mathcal{T} \subsetneq \mathcal{T}_k(x)$ be the largest subcollection with dual feasible support: Algorithm 3 goes through to Step 5. and updates x to $x + \alpha d$. Then we have $\Theta(E(x + \alpha d)) <_L \Theta(E(x))$. Furthermore,

(a) if $\mathcal{T} = \emptyset$, then $\varepsilon_k(x + \alpha d) < \varepsilon_k(x)$,

(b) if $\mathcal{T} \neq \emptyset$, then $\varepsilon_k(x + \alpha d) = \varepsilon_k(x) = \varepsilon_k(\nu)$ and $\mathcal{T}_k(x + \alpha d) = \mathcal{T}_k(\nu)$.

Proof: $x + \alpha d \in pi(v)$ if $x \in pi(v)$

- $x(S) = (x + \alpha d)(S)$ for all $S \in \text{span}(\mathcal{R}_{k-1})$
- $\Theta(E(x))$ and $\Theta(E(x + \alpha d))$ both starts with the same values $E(S, \nu) \geq \varepsilon_{k-1}(\nu)$
- focus on truncated excess vectors: $2^N \setminus \text{span}(\mathcal{R}_{k-1})$

(a): next value in both corresponds to some $S \in \mathcal{T}_k$

- since $d(S) \geq 1$ and $\alpha > 0$: $\varepsilon_k(x + \alpha d) < \varepsilon_k(x)$ (Fig. 4)

Constructive algorithm

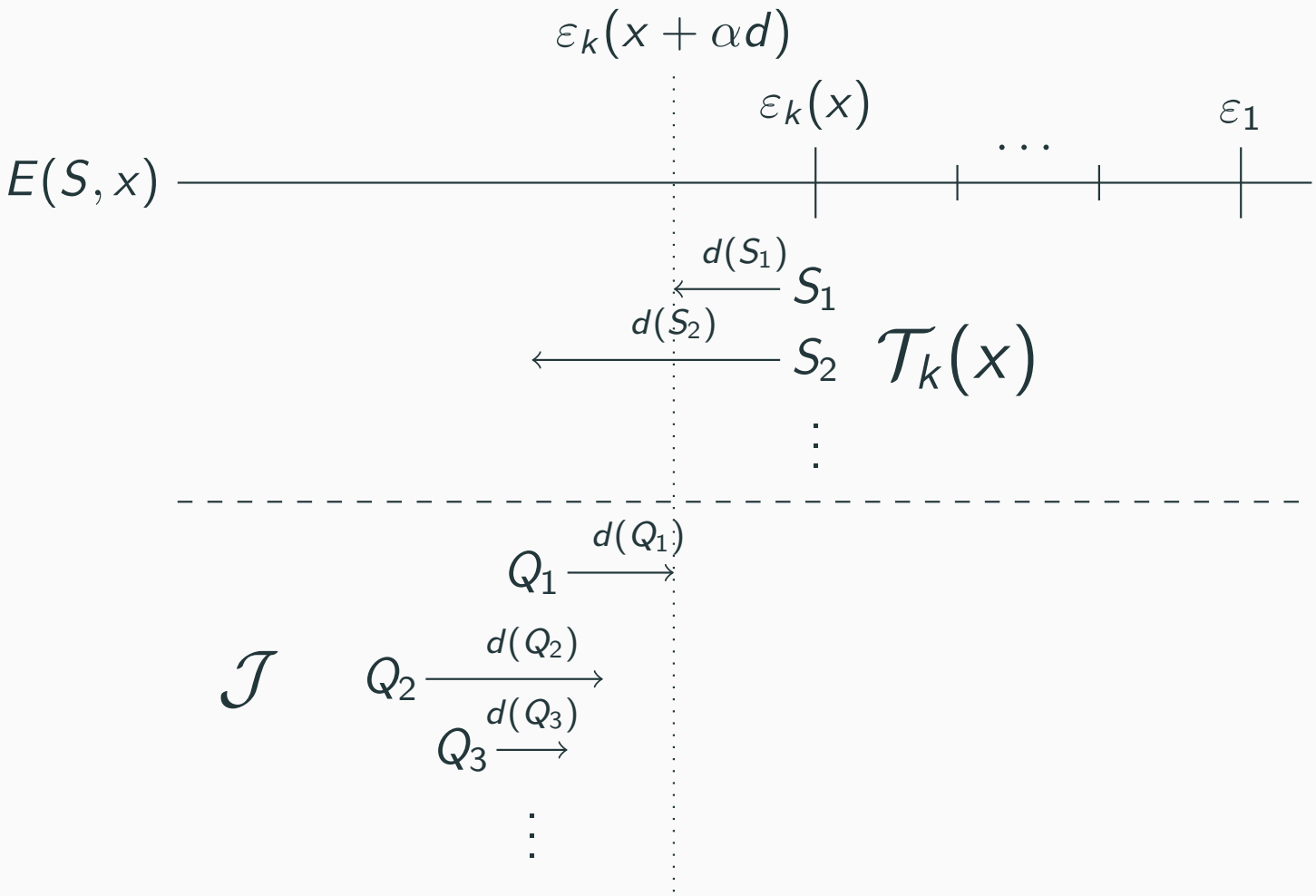


Figure 4: Optimal step size

Lexicographical descent

Lemma 2

Let $\mathcal{T} \subsetneq \mathcal{T}_k(x)$ be the largest subcollection with dual feasible support: Algorithm 3 goes through to Step 5. and updates x to $x + \alpha d$. Then we have $\Theta(E(x + \alpha d)) <_L \Theta(E(x))$. Furthermore,

(a) if $\mathcal{T} = \emptyset$, then $\varepsilon_k(x + \alpha d) < \varepsilon_k(x)$,

(b) if $\mathcal{T} \neq \emptyset$, then $\varepsilon_k(x + \alpha d) = \varepsilon_k(x) = \varepsilon_k(\nu)$ and $\mathcal{T}_k(x + \alpha d) = \mathcal{T}_k(\nu)$.

Proof: $x + \alpha d \in pi(v)$ if $x \in pi(v)$

- $x(S) = (x + \alpha d)(S)$ for all $S \in \text{span}(\mathcal{R}_{k-1})$
- $\Theta(E(x))$ and $\Theta(E(x + \alpha d))$ both starts with the same values $E(S, \nu) \geq \varepsilon_{k-1}(\nu)$
- focus on truncated excess vectors: $2^N \setminus \text{span}(\mathcal{R}_{k-1})$

(a): next value in both corresponds to some $S \in \mathcal{T}_k$

- since $d(S) \geq 1$ and $\alpha > 0$: $\varepsilon_k(x + \alpha d) < \varepsilon_k(x)$ (Fig. 4)
- hence $\Theta(E(x + \alpha d)) <_L \Theta(E(x))$

Constructive algorithm

(b): $d(Q) = 0$ for all $Q \in \mathcal{T}$

Constructive algorithm

(b): $d(Q) = 0$ for all $Q \in \mathcal{T}$

- since $\exists Q \in \mathcal{T}$: $\varepsilon_k(x + \alpha d) = \varepsilon_k(x) = E(Q, x)$ for $Q \in \mathcal{T}$

Constructive algorithm

(b): $d(Q) = 0$ for all $Q \in \mathcal{T}$

- since $\exists Q \in \mathcal{T}$: $\varepsilon_k(x + \alpha d) = \varepsilon_k(x) = E(Q, x)$ for $Q \in \mathcal{T}$
- then we have $\mathcal{T} = \mathcal{T}_k(x + \alpha d) \subsetneq \mathcal{T}_k(x)$ (Fig. 5)

Constructive algorithm

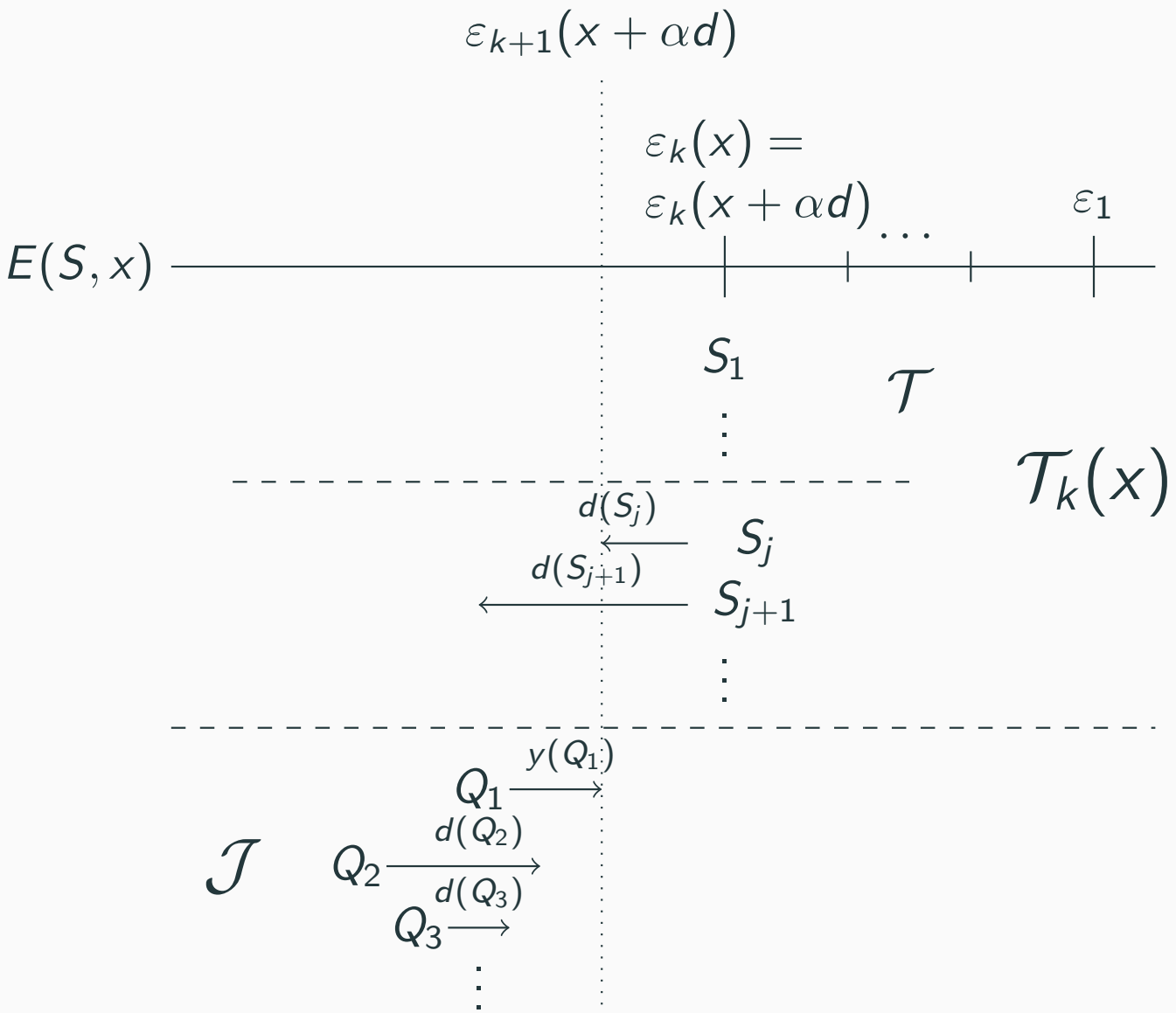


Figure 5: Step when $\mathcal{T} \neq \emptyset$

Constructive algorithm

(b): $d(Q) = 0$ for all $Q \in \mathcal{T}$

- since $\exists Q \in \mathcal{T}$: $\varepsilon_k(x + \alpha d) = \varepsilon_k(x) = E(Q, x)$ for $Q \in \mathcal{T}$
- then we have $\mathcal{T} \cap \mathcal{T}_k(x) = \mathcal{T}_k(x + \alpha d) \subsetneq \mathcal{T}_k(x)$ (Fig. 5)
- then the truncated excess vector starts with *less* $\varepsilon_k(x)$ values because of $|\mathcal{T}_k(x + \alpha d)| < |\mathcal{T}_k(x)|$

Constructive algorithm

(b): $d(Q) = 0$ for all $Q \in \mathcal{T}$

- since $\exists Q \in \mathcal{T}$: $\varepsilon_k(x + \alpha d) = \varepsilon_k(x) = E(Q, x)$ for $Q \in \mathcal{T}$
- then we have $\mathcal{T} \cap \mathcal{T}_k(x) = \mathcal{T}_k(x + \alpha d) \subsetneq \mathcal{T}_k(x)$ (Fig. 5)
- then the truncated excess vector starts with *less* $\varepsilon_k(x)$ values because of $|\mathcal{T}_k(x + \alpha d)| < |\mathcal{T}_k(x)|$
- Consequently $\Theta(E(x + \alpha d)) <_L \Theta(E(x))$

Constructive algorithm

(b): $d(Q) = 0$ for all $Q \in \mathcal{T}$

- since $\exists Q \in \mathcal{T}$: $\varepsilon_k(x + \alpha d) = \varepsilon_k(x) = E(Q, x)$ for $Q \in \mathcal{T}$
- then we have $\mathcal{T} \cap \mathcal{T}_k(x) = \mathcal{T}_k(x + \alpha d) \subsetneq \mathcal{T}_k(x)$ (Fig. 5)
- then the truncated excess vector starts with *less* $\varepsilon_k(x)$ values because of $|\mathcal{T}_k(x + \alpha d)| < |\mathcal{T}_k(x)|$
- Consequently $\Theta(E(x + \alpha d)) <_L \Theta(E(x))$
- Furthermore, since $\mathcal{T}_k(x + \alpha d)$ is *balanced*, we have $\varepsilon_k(x + \alpha d) = \varepsilon_k(\nu)$ and $\mathcal{T}_k(x + \alpha d) = \mathcal{T}_k(\nu)$ $\square$

Constructive algorithm

Main theorem

Algorithm 3 stops after finite steps. The **while** loop executes at most $(n-1)$ iterations before finding ν

Constructive algorithm

Main theorem

Algorithm 3 stops after finite steps. The **while** loop executes at most $(n-1)$ iterations before finding ν

Proof: $\text{rank}(\mathcal{R}^{(k-1)})$ increases in every iteration:

Constructive algorithm

Main theorem

Algorithm 3 stops after finite steps. The **while** loop executes at most $(n-1)$ iterations before finding ν

Proof: $\text{rank}(\mathcal{R}^{(k-1)})$ increases in every iteration:

- $\emptyset \neq \mathcal{T}_k \subseteq 2^N \setminus \text{span}(\mathcal{R}^{(k-1)})$, $\text{rank}(\mathcal{R}^{(0)}) = 1 \Rightarrow$ at most $(n-1)$ iterations

Constructive algorithm

Main theorem

Algorithm 3 stops after finite steps. The **while** loop executes at most $(n-1)$ iterations before finding ν

Proof: $\operatorname{rank}(\mathcal{R}^{(k-1)})$ increases in every iteration:

- $\emptyset \neq \mathcal{T}_k \subseteq 2^N \setminus \operatorname{span}(\mathcal{R}^{(k-1)})$, $\operatorname{rank}(\mathcal{R}^{(0)}) = 1 \Rightarrow$ at most $(n-1)$ iterations
- enough: within a single iteration finite x to $x + \alpha d$ pivot steps leads to *balanced* tight set

Constructive algorithm

Main theorem

Algorithm 3 stops after finite steps. The **while** loop executes at most $(n-1)$ iterations before finding ν

Proof: $\operatorname{rank}(\mathcal{R}^{(k-1)})$ increases in every iteration:

- $\emptyset \neq \mathcal{T}_k \subseteq 2^N \setminus \operatorname{span}(\mathcal{R}^{(k-1)})$, $\operatorname{rank}(\mathcal{R}^{(0)}) = 1 \Rightarrow$ at most $(n-1)$ iterations
- enough: within a single iteration finite x to $x + \alpha d$ pivot steps leads to *balanced* tight set
- Lemma 2: as soon as $\mathcal{T} \neq \emptyset$, we found a *balanced* tight set $\Rightarrow$ assume $\mathcal{T} = \emptyset$

Constructive algorithm

Main theorem

Algorithm 3 stops after finite steps. The **while** loop executes at most $(n-1)$ iterations before finding ν

Proof: $\operatorname{rank}(\mathcal{R}^{(k-1)})$ increases in every iteration:

- $\emptyset \neq \mathcal{T}_k \subseteq 2^N \setminus \operatorname{span}(\mathcal{R}^{(k-1)})$, $\operatorname{rank}(\mathcal{R}^{(0)}) = 1 \Rightarrow$ at most $(n-1)$ iterations
- enough: within a single iteration finite x to $x + \alpha d$ pivot steps leads to *balanced* tight set
- Lemma 2: as soon as $\mathcal{T} \neq \emptyset$, we found a *balanced* tight set $\Rightarrow$ assume $\mathcal{T} = \emptyset$
- possible tight sets are finite: $\mathcal{T}_k \subseteq 2^N \setminus \operatorname{span}(\mathcal{R}_{k-1})$, among which there are *balanced* as well (e.g. $\mathcal{T}_k(\nu)$)

Constructive algorithm

Main theorem

Algorithm 3 stops after finite steps. The **while** loop executes at most $(n-1)$ iterations before finding ν

Proof: $\operatorname{rank}(\mathcal{R}^{(k-1)})$ increases in every iteration:

- $\emptyset \neq \mathcal{T}_k \subseteq 2^N \setminus \operatorname{span}(\mathcal{R}^{(k-1)})$, $\operatorname{rank}(\mathcal{R}^{(0)}) = 1 \Rightarrow$ at most $(n-1)$ iterations
- enough: within a single iteration finite x to $x + \alpha d$ pivot steps leads to *balanced* tight set
- Lemma 2: as soon as $\mathcal{T} \neq \emptyset$, we found a *balanced* tight set $\Rightarrow$ assume $\mathcal{T} = \emptyset$
- possible tight sets are finite: $\mathcal{T}_k \subseteq 2^N \setminus \operatorname{span}(\mathcal{R}_{k-1})$, among which there are *balanced* as well (e.g. $\mathcal{T}_k(\nu)$)
- suppose we find an infinite series of *non-balanced* tight sets $T = (\mathcal{T}^1, \dots)$

Constructive algorithm

Main theorem

Algorithm 3 stops after finite steps. The **while** loop executes at most $(n-1)$ iterations before finding ν

Proof: $\operatorname{rank}(\mathcal{R}^{(k-1)})$ increases in every iteration:

- $\emptyset \neq \mathcal{T}_k \subseteq 2^N \setminus \operatorname{span}(\mathcal{R}^{(k-1)})$, $\operatorname{rank}(\mathcal{R}^{(0)}) = 1 \Rightarrow$ at most $(n-1)$ iterations
- enough: within a single iteration finite x to $x + \alpha d$ pivot steps leads to *balanced* tight set
- Lemma 2: as soon as $\mathcal{T} \neq \emptyset$, we found a *balanced* tight set $\Rightarrow$ assume $\mathcal{T} = \emptyset$
- possible tight sets are finite: $\mathcal{T}_k \subseteq 2^N \setminus \operatorname{span}(\mathcal{R}_{k-1})$, among which there are *balanced* as well (e.g. $\mathcal{T}_k(\nu)$)
- suppose we find an infinite series of *non-balanced* tight sets $T = (\mathcal{T}^1, \dots)$
- finitely many tight sets: we must visit (at least) one infinitely many times, denote it by $\mathcal{T}^1$

Constructive algorithm

- w.l.o.g. truncate the series only containing tight sets that we visit infinitely many times

Constructive algorithm

- w.l.o.g. truncate the series only containing tight sets that we visit infinitely many times
- $T = (\mathcal{T}^1, \mathcal{T}^2, \dots, \mathcal{T}^m, \mathcal{T}^1, \mathcal{T}^{m+1}, \dots)$

Constructive algorithm

- w.l.o.g. truncate the series only containing tight sets that we visit infinitely many times
- $T = (\mathcal{T}^1, \mathcal{T}^2, \dots, \mathcal{T}^m, \mathcal{T}^1, \mathcal{T}^{m+1}, \dots)$
- even though upon revisiting $\mathcal{T}^1$ the LP (impr-dir) is the same, the solution might not be unique $\Rightarrow \mathcal{T}^2 \neq \mathcal{T}^{m+1}$ necessarily

Constructive algorithm

- w.l.o.g. truncate the series only containing tight sets that we visit infinitely many times
- $T = (\mathcal{T}^1, \mathcal{T}^2, \dots, \mathcal{T}^m, \mathcal{T}^1, \mathcal{T}^{m+1}, \dots)$
- even though upon revisiting $\mathcal{T}^1$ the LP (impr-dir) is the same, the solution might not be unique $\Rightarrow \mathcal{T}^2 \neq \mathcal{T}^{m+1}$ necessarily
- w.l.o.g. we can choose m such that $\mathcal{T}^2 = \mathcal{T}^{m+1}$

Constructive algorithm

- w.l.o.g. truncate the series only containing tight sets that we visit infinitely many times
- $T = (\mathcal{T}^1, \mathcal{T}^2, \dots, \mathcal{T}^m, \mathcal{T}^1, \mathcal{T}^{m+1}, \dots)$
- even though upon revisiting $\mathcal{T}^1$ the LP (impr-dir) is the same, the solution might not be unique $\Rightarrow \mathcal{T}^2 \neq \mathcal{T}^{m+1}$ necessarily
- w.l.o.g. we can choose m such that $\mathcal{T}^2 = \mathcal{T}^{m+1}$
- in fact we can choose m such that in T *precisely* $\mathcal{T}^1, \mathcal{T}^2, \dots, \mathcal{T}^m$ is repeated infinitely many times

Constructive algorithm

- w.l.o.g. truncate the series only containing tight sets that we visit infinitely many times
- $T = (\mathcal{T}^1, \mathcal{T}^2, \dots, \mathcal{T}^m, \mathcal{T}^1, \mathcal{T}^{m+1}, \dots)$
- even though upon revisiting $\mathcal{T}^1$ the LP (impr-dir) is the same, the solution might not be unique $\Rightarrow \mathcal{T}^2 \neq \mathcal{T}^{m+1}$ necessarily
- w.l.o.g. we can choose m such that $\mathcal{T}^2 = \mathcal{T}^{m+1}$
- in fact we can choose m such that in T *precisely* $\mathcal{T}^1, \mathcal{T}^2, \dots, \mathcal{T}^m$ is repeated infinitely many times
- let us denote with $(d^1, d^2, \dots)$, $(\alpha^1, \alpha^2, \dots)$, $(\varepsilon^1, \varepsilon^2, \dots, \varepsilon^m, \varepsilon^{m+1})$

Constructive algorithm

- w.l.o.g. truncate the series only containing tight sets that we visit infinitely many times
- $T = (\mathcal{T}^1, \mathcal{T}^2, \dots, \mathcal{T}^m, \mathcal{T}^1, \mathcal{T}^{m+1}, \dots)$
- even though upon revisiting $\mathcal{T}^1$ the LP (impr-dir) is the same, the solution might not be unique $\Rightarrow \mathcal{T}^2 \neq \mathcal{T}^{m+1}$ necessarily
- w.l.o.g. we can choose m such that $\mathcal{T}^2 = \mathcal{T}^{m+1}$
- in fact we can choose m such that in T *precisely* $\mathcal{T}^1, \mathcal{T}^2, \dots, \mathcal{T}^m$ is repeated infinitely many times
- let us denote with $(d^1, d^2, \dots)$, $(\alpha^1, \alpha^2, \dots)$, $(\varepsilon^1, \varepsilon^2, \dots, \varepsilon^m, \varepsilon^{m+1})$
- by Lemma 2 we have $\varepsilon^1 > \varepsilon^m > \varepsilon^{m+1}$ and $x + \sum_{j=1}^{m} \alpha^j d^j \in pi(v)$

Constructive algorithm

- w.l.o.g. truncate the series only containing tight sets that we visit infinitely many times
- $T = (\mathcal{T}^1, \mathcal{T}^2, \dots, \mathcal{T}^m, \mathcal{T}^1, \mathcal{T}^{m+1}, \dots)$
- even though upon revisiting $\mathcal{T}^1$ the LP (impr-dir) is the same, the solution might not be unique $\Rightarrow \mathcal{T}^2 \neq \mathcal{T}^{m+1}$ necessarily
- w.l.o.g. we can choose m such that $\mathcal{T}^2 = \mathcal{T}^{m+1}$
- in fact we can choose m such that in T *precisely* $\mathcal{T}^1, \mathcal{T}^2, \dots, \mathcal{T}^m$ is repeated infinitely many times
- let us denote with $(d^1, d^2, \dots)$, $(\alpha^1, \alpha^2, \dots)$, $(\varepsilon^1, \varepsilon^2, \dots, \varepsilon^m, \varepsilon^{m+1})$
- by Lemma 2 we have $\varepsilon^1 > \varepsilon^m > \varepsilon^{m+1}$ and $x + \sum_{j=1}^{m} \alpha^j d^j \in pi(v)$ while d^1 is the solution of

$$\begin{aligned} \min_{d} \quad & \sum_{S \in \mathcal{T}^1} d(S) & & \\ s.t. \quad & d(S) & \geq 1 & \quad \forall S \in \mathcal{T}^1 \\ & d(P) & = 0 & \quad \forall P \in \mathcal{R}_{k-1} \end{aligned}$$

Constructive algorithm

- $\sum_{j=1}^{m} \alpha^j d^j(S) = \varepsilon^1 - \varepsilon^{m+1}$ is constant through $\mathcal{T}^1$

Constructive algorithm

- $\sum_{j=1}^{m} \alpha^j d^j(S) = \varepsilon^1 - \varepsilon^{m+1}$ is constant through $\mathcal{T}^1$
- $\Rightarrow \beta \sum_{j=1}^{m} \alpha^j d^j$ is feasible for large enough $\beta > 0$

Constructive algorithm

- $\sum_{j=1}^{m} \alpha^j d^j(S) = \varepsilon^1 - \varepsilon^{m+1}$ is constant through $\mathcal{T}^1$
- $\Rightarrow \beta \sum_{j=1}^{m} \alpha^j d^j$ is feasible for large enough $\beta > 0$
- $\Rightarrow \frac{\sum_{j=1}^{m} \alpha^j d^j}{\varepsilon^1 - \varepsilon^{m+1}}$ is optimal

Constructive algorithm

- $\sum_{j=1}^{m} \alpha^j d^j(S) = \varepsilon^1 - \varepsilon^{m+1}$ is constant through $\mathcal{T}^1$
- $\Rightarrow \beta \sum_{j=1}^{m} \alpha^j d^j$ is feasible for large enough $\beta > 0$
- $\Rightarrow \frac{\sum_{j=1}^{m} \alpha^j d^j}{\varepsilon^1 - \varepsilon^{m+1}}$ is optimal

$\Rightarrow$ $d^1(S) = 1$ for all $S \in \mathcal{T}^1$ as d^1 is also optimal

Constructive algorithm

- $\sum_{j=1}^{m} \alpha^j d^j(S) = \varepsilon^1 - \varepsilon^{m+1}$ is constant through $\mathcal{T}^1$
- $\Rightarrow \beta \sum_{j=1}^{m} \alpha^j d^j$ is feasible for large enough $\beta > 0$
- $\Rightarrow \frac{\sum_{j=1}^{m} \alpha^j d^j}{\varepsilon^1 - \varepsilon^{m+1}}$ is optimal

$\Rightarrow$ $d^1(S) = 1$ for all $S \in \mathcal{T}^1$ as d^1 is also optimal

$\Rightarrow$ all coalitions remain tight, while some join $\Rightarrow \mathcal{T}^1 \subsetneq \mathcal{T}^2$ (Fig. 6)

Constructive algorithm

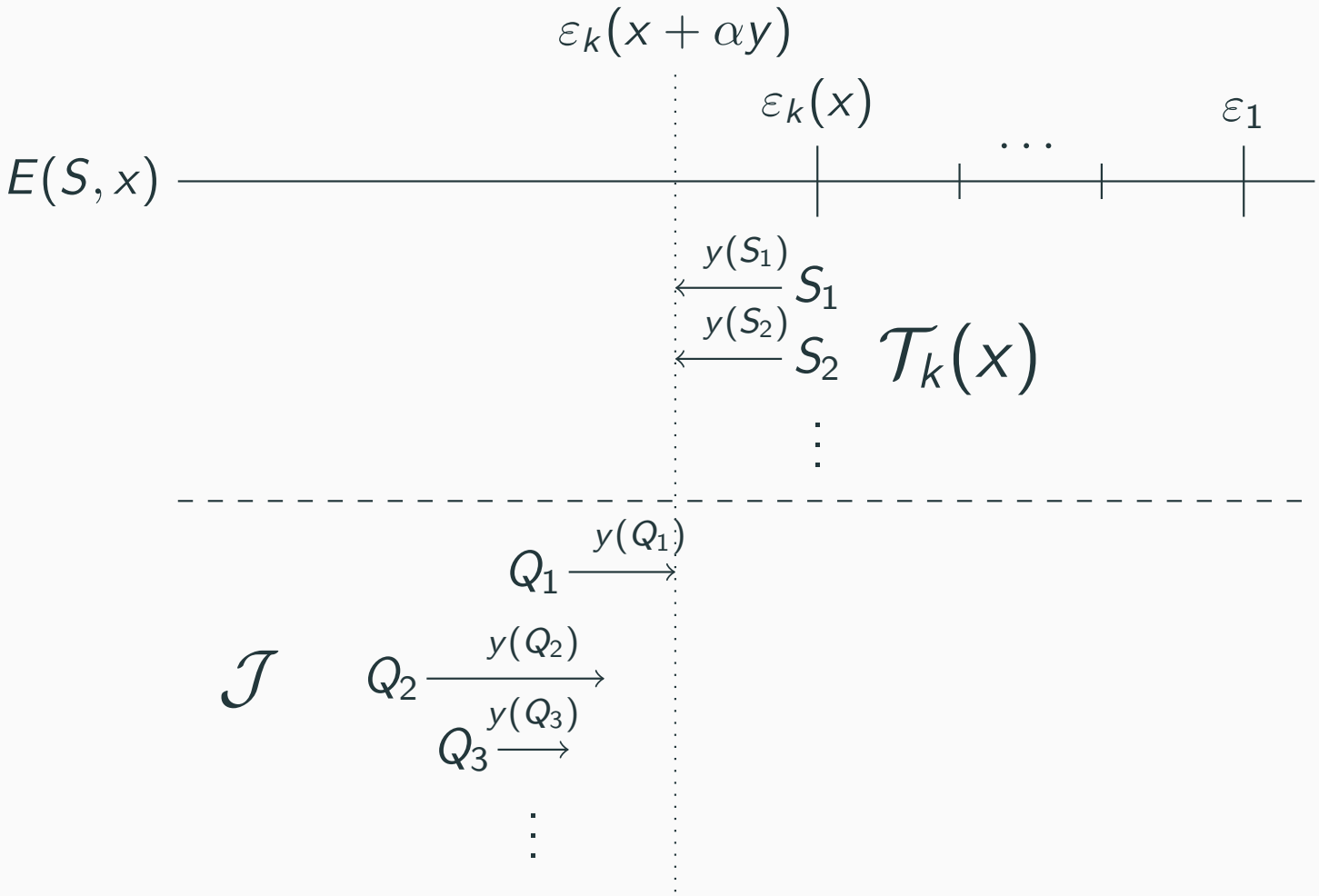


Figure 6: Optimal step size

Constructive algorithm

- $\sum_{j=1}^{m} \alpha^j d^j(S) = \varepsilon^1 - \varepsilon^{m+1}$ is constant through $\mathcal{T}^1$
- $\Rightarrow \beta \sum_{j=1}^{m} \alpha^j d^j$ is feasible for large enough $\beta > 0$
- $\Rightarrow \frac{\sum_{j=1}^{m} \alpha^j d^j}{\varepsilon^1 - \varepsilon^{m+1}}$ is optimal

$\Rightarrow$ $d^1(S) = 1$ for all $S \in \mathcal{T}^1$ as d^1 is also optimal

$\Rightarrow$ all coalitions remain tight, while some join $\Rightarrow \mathcal{T}^1 \subsetneq \mathcal{T}^2$ (Fig. 6)

$\Rightarrow$ repeating throughout the series leads to $\mathcal{T}^1 \subsetneq \mathcal{T}^1$ $\square$

Type I and II games

Type I and II games both appear in [Potters et al., 1996] and [Reijnierse, 1995]. The characteristic function for type I is given by

$$v(S) = \begin{cases} 0, & \text{if } |S| = 1 \\ \text{random integer between 1 and } 100|S|, & \text{if } 2 \leq |S| < n \\ \text{random integer between } 100(n-2) \text{ and } 100n, & \text{if } S = N. \end{cases}$$

while type II games are generated as

$$v(S) = \begin{cases} 0, & \text{if } |S| = 1 \\ \text{random integer between 1 and } 50n, & \text{otherwise.} \end{cases}$$

Type III and IV games

[Derks and Kuipers, 1997] were looking for games where the number of iterations grow more or less linearly with the number of players, and so they introduced type III games as

$$v(S) = \begin{cases} 0, & \text{if } |S| < n-2 \\ 1 \text{ with probability } 0.9, & \text{if } n-2 \leq |S| < n \\ 1, & \text{if } S = S. \end{cases}$$

In order to test the methods on games which are more realistically distinguish between the number of iterations required by primal and dual methods we introduce type IV games as

$$v(S) = \begin{cases} 0, & \text{if } |S| = 1 \\ \text{random integer between 1 and } n, & \text{otherwise.} \end{cases}$$

UN Security Council game

We as well attempt to find games that our method *LD* struggles with, so we adopted the United Nations (UN) Security Council voting mechanism into weighted voting games with arbitrary size, where there are 5 *big* (veto) players and the rest (originally 10) are *small*.

Weights are calibrated such that all 5 veto players' agreement needed to pass a vote, while still approx. half of the small players also needed. The weights are for $n \geq 7$: $w_j = \lfloor \frac{n-3}{2} \rfloor$ for $j = 1, \ldots, 5$ and $w_j = 1$ otherwise, the quota is $q = 4w_1 + n - 4$, and the game is

$$v(S) = \begin{cases} 1, & \text{if } w(S) \geq q \\ 0, & \text{otherwise.} \end{cases}$$

Computational results

Table 4: Type I and II games

n	Time				Iterations				Pivots				Subrout.	
	LD	DK	P	D	LD	DK	P	D	LD	DK	P	D	LD	P
15	0.10	0.17	0.42	0.17	1.50	1.52	1.50	1.52	19.2	25.9	93.6	153	21.0	2.0
20	4.40	7.69	22.1	8.43	1.46	1.46	1.46	1.46	28.9	40.6	159	330	31.4	1.9
25	227	425	OoM	OoT	1.78	1.8	OoM	OoT	42.1	56.5	OoM	OoT	47.6	OoM
26	463	958	-	-	1.6	1.6	-	-	40.6	71.8	-	-	43.3	-
27	1261	2047	-	-	1.9	1.9	-	-	57.1	70.4	-	-	69.0	-
28	4406	6421	-	-	1.9	1.9	-	-	48.5	82.1	-	-	53.7	-
29	12220	17796	-	-	1.7	1.7	-	-	58.1	78.0	-	-	66.4	-
30	19232	OoT	-	-	1.4	OoT	-	-	44.4	OoT	-	-	47.4	-
15	0.13	0.18	0.62	0.29	2.5	2.5	2.5	2.6	21.6	21.0	116	103	26.4	4
20	5.06	9.05	38.0	15.0	2.5	2.5	2.5	2.5	34.1	34.0	257	198	42.5	4
25	294	611	OoM	OoT	3.6	3.6	OoM	OoT	53.2	69.0	OoM	OoT	69.6	OoM
26	524	1155	-	-	3.1	3.1	-	-	42.7	65.6	-	-	49.0	-
27	1167	2322	-	-	3.0	3.1	-	-	52.3	69.1	-	-	66.3	-
28	5222	7846	-	-	4.8	4.8	-	-	58.3	61.7	-	-	77.6	-
29	14624	21429	-	-	4.4	4.4	-	-	69.7	78.3	-	-	96.7	-
30	29593	OoT	-	-	3.8	OoT	-	-	68.1	OoT	-	-	90.7	-

Computational results

Table 5: Type III and V games

n	Time				Iterations				Pivots				Subrout.	
	LD	DK	P	D	LD	DK	P	D	LD	DK	P	D	LD	P
15	0.007	0.73	0.18	0.25	1	6.7	1	2.6	0	110	23.5	102	3.9	2.4
20	0.18	47.6	8.44	15.2	1	11.0	1	3.0	0	234	33.8	933	3.8	2.8
25	6.48	2571	OoM	OoT	1	11.6	OoM	OoT	0	348	OoM	OoT	4.7	OoM
26	13.6	5705	-	-	1	13.5	-	-	0	385	-	-	5.1	-
27	29.1	12208	-	-	1	12.6	-	-	0	408	-	-	5.2	-
28	729	29867	-	-	1	14.7	-	-	0	471	-	-	5.4	-
29	120	OoT	-	-	1	OoT	-	-	0	OoT	-	-	6	-
30	239	-	-	-	1	-	-	-	0	-	-	-	6	-
10	0.007	0.046	0.016	0.21	2	6	2	3	2	34	18	47	5	6
15	0.11	0.45	0.46	0.41	2	7	2	5	2	60	26	144	5	4
20	4.55	21.9	21.9	12.6	2	4	2	3	2	155	33	156	6	5
25	174	2332	OoM	OoT	2	12	OoM	OoT	2	458	OoM	OoT	5	OoM
26	1164	13348	-	-	2	10	-	-	2	1328	-	-	5	-
27	16191	6506	-	-	2	15	-	-	2	260	-	-	5	-
28	OoT	10269	-	-	OoT	10	-	-	OoT	148	-	-	OoT	-
29	-	OoT	-	-	-	OoT	-	-	-	OoT	-	-	-	-