Optimalizace & soft omezení: modely (dokončení)

Omezení nad polookruhy

- *Semiring-based CSP*
- c-**polookruh** $\langle \mathcal{A}, +, \times, \mathbf{0}, \mathbf{1} \rangle$
  - $\mathcal{A}$ množina polookruhu (množina preferencí)
  - $\mathbf{0} \in \mathcal{A}$ (úplné nesplnění), $\mathbf{1} \in \mathcal{A}$ (úplné splnění)
  - $+$ komutativní, idempotentní, asociativní operace

    s jednotkovým prvkem $\mathbf{0}$, s absorbujím prvkem $\mathbf{1}$
  - $\times$ komutativní, asociativní operace, distributivní nad $+$,

    s jednotkovým prvkem $\mathbf{1}$, s absorbujím prvkem $\mathbf{0}$
- **$\times$ se používá ke kombinaci preferencí několika omezení** $\min$ u fuzzy CSP

  $\mathbf{0}$ minimum (nesplnění), $\mathbf{1}$ maximum (splnění)
- **Částečné uspořádání $\leq_S$**: $a \leq_S b$ právě tehdy, když $a + b = b$

  používá se k výběru „lepšího” přiřazení $\max$ u fuzzy CSP

Instance omezení nad polookruhy

Přístup	$\mathcal{A}$	$+$ uspořádání	$\times$ kombinace	$\mathbf{0}$	$\mathbf{1}$
CSP	$\{0, 1\}$	$\vee$	$\wedge$	0	1

Instance omezení nad polookruhy

Přístup	$\mathcal{A}$	$+$ uspořádání	$\times$ kombinace	$\mathbf{0}$	$\mathbf{1}$
CSP	$\{0, 1\}$	$\vee$	$\wedge$	0	1
Fuzzy CSP	$\langle 0, 1 \rangle$	$\max$	$\min$	0	1

Instance omezení nad polookruhy

Přístup	$\mathcal{A}$	$+$ uspořádání	$\times$ kombinace	$\mathbf{0}$	$\mathbf{1}$
CSP	$\{0,1\}$	$\vee$	$\wedge$	0	1
Fuzzy CSP	$\langle 0,1 \rangle$	$\max$	$\min$	0	1
CSP s váhami	$\mathbb{N} \cup \{+\infty\}$	$\min$	$+$	$+\infty$	0

Instance omezení nad polookruhy

Přístup	$\mathcal{A}$	$+$	$\times$	$\mathbf{0}$	$\mathbf{1}$
		uspořádání	kombinace		
CSP	$\{0,1\}$	$\vee$	$\wedge$	0	1
Fuzzy CSP	$\langle 0,1\rangle$	$\max$	$\min$	0	1
CSP s váhami	$\mathbb{N}\cup\{+\infty\}$	$\min$	$+$	$+\infty$	0

- Důležité vlastnosti:
  - **striktní monotonie**: $\forall a,b,c \in \mathcal{A} : ((a < c) \wedge (b \neq \mathbf{0})) \Rightarrow ((a \times b) < (c \times b))$

    fakt, že něco lze lokálně zlepšit nelze globálně ignorovat (platí pro CSP s váhami)

    př. $a = 0.3, c = 0.4, b = 0.2$ pro fuzzy CSP: není striktní monotonie

Instance omezení nad polookruhy

Přístup	$\mathcal{A}$	$+$ uspořádání	$\times$ kombinace	$\mathbf{0}$	$\mathbf{1}$
CSP	$\{0,1\}$	$\vee$	$\wedge$	0	1
Fuzzy CSP	$\langle 0,1\rangle$	$\max$	$\min$	0	1
CSP s váhami	$\mathbb{N} \cup \{+\infty\}$	$\min$	$+$	$+\infty$	0

- Důležité vlastnosti:
  - striktní monotonie: $\forall a,b,c \in \mathcal{A} : ((a < c) \wedge (b \neq \mathbf{0})) \Rightarrow ((a \times b) < (c \times b))$
    fakt, že něco lze lokálně zlepšit nelze globálně ignorovat (platí pro CSP s váhami)
    př. $a = 0.3, c = 0.4, b = 0.2$ pro fuzzy CSP: není striktní monotonie
  - idempotence: $\forall a \in \mathcal{A} : a \times a = a$ (platí pro fuzzy CSP)
    omezení, které je v problému obsaženo, může být do něj přidáno beze změny významu
    př. $x > 1@10, x > 1@10, x = 0@\infty$, přiřazení $x = 0$ má pro CSP s váhami úroveň konz. 20

Instance omezení nad polookruhy

Přístup	$\mathcal{A}$	$+$ uspořádání	$\times$ kombinace	$\mathbf{0}$	$\mathbf{1}$
CSP	$\{0,1\}$	$\vee$	$\wedge$	0	1
Fuzzy CSP	$\langle 0,1\rangle$	$\max$	$\min$	0	1
CSP s váhami	$\mathbb{N}\cup\{+\infty\}$	$\min$	$+$	$+\infty$	0

- Důležité vlastnosti:
  - **striktní monotonie**: $\forall a,b,c \in \mathcal{A} : ((a<c) \wedge (b \neq \mathbf{0})) \Rightarrow ((a \times b) < (c \times b))$
    fakt, že něco lze lokálně zlepšit nelze globálně ignorovat (platí pro CSP s váhami)
    př. $a = 0.3, c = 0.4, b = 0.2$ pro fuzzy CSP: není striktní monotonie
  - **idempotence**: $\forall a \in \mathcal{A} : a \times a = a$ (platí pro fuzzy CSP)
    omezení, které je v problému obsaženo, může být do něj přidáno beze změny významu
    př. $x > 1@10, x > 1@10, x = 0@\infty$, přiřazení $x = 0$ má pro CSP s váhami úroveň konz. 20
  - striktní monotonie a idempotence $\times$ zároveň pouze pro dvouprvkové $\mathcal{A}$, tj. jen pro CSP

Instance omezení nad polookruhy

Přístup	$\mathcal{A}$	$+$ uspořádání	$\times$ kombinace	$\mathbf{0}$	$\mathbf{1}$
CSP	$\{0,1\}$	$\vee$	$\wedge$	0	1
Fuzzy CSP	$\langle 0,1\rangle$	max	min	0	1
CSP s váhami	$\mathbb{N}\cup\{+\infty\}$	min	$+$	$+\infty$	0

- Důležité vlastnosti:
  - **striktní monotonie**: $\forall a,b,c \in \mathcal{A} : ((a < c) \wedge (b \neq \mathbf{0})) \Rightarrow ((a \times b) < (c \times b))$
    fakt, že něco lze lokálně zlepšit nelze globálně ignorovat (platí pro CSP s váhami)
    př. $a = 0.3, c = 0.4, b = 0.2$ pro fuzzy CSP: není striktní monotonie
  - **idempotence**: $\forall a \in \mathcal{A} : a \times a = a$ (platí pro fuzzy CSP)
    omezení, které je v problému obsaženo, může být do něj přidáno beze změny významu
    př. $x > 1@10, x > 1@10, x = 0@\infty$, přiřazení $x = 0$ má pro CSP s váhami úroveň konz. 20
  - striktní monotonie a idempotence $\times$ zároveň pouze pro dvouprvkové $\mathcal{A}$, tj. jen pro CSP
- Existující vztahy: CSP $\equiv$ fuzzy CSP na dvouprvkové $\mathcal{A}$
  fuzzy CSP lze polynomiálně transformovat na CSP s váhami

Definice omezení nad polookruhy

- **Systém** (S, D, V):

  S c-polookruh, D konečná doména, V uspořádaná množina proměnných

- **Soft omezení** (def, con): rozsah omezení $con \subseteq V$, $def : D^{|con|} \rightarrow \mathcal{A}$

- **Soft problém** je (C, con) nad (S, D, V), kde $con \subseteq V$, C množina omezení

Definice omezení nad polookruhy

- **Systém** (S, D, V):

  S c-polookruh, D konečná doména, V uspořádaná množina proměnných

- **Soft omezení** (def, con): rozsah omezení $con \subseteq V$, $def : D^{|con|} \rightarrow \mathcal{A}$

- **Soft problém** je (C, con) nad (S, D, V), kde $con \subseteq V$, C množina omezení

- **Projekce n-tic** $\vec{t} \downarrow^{X}_{Y}$

  - **příklad:** $(1, 2, 3, 4, 5) \downarrow^{\text{(A,B,C,D,E)}}_{\text{(D,A,E)}} = (4, 1, 5)$

Definice omezení nad polookruhy

- **Systém** (S, D, V):

  S c-polookruh, D konečná doména, V uspořádaná množina proměnných

- **Soft omezení** (def, con): rozsah omezení $con \subseteq V$, $def : D^{|con|} \rightarrow \mathcal{A}$

- **Soft problém** je (C, con) nad (S, D, V), kde $con \subseteq V$, C množina omezení

- **Projekce n-tic** $\vec{t} \downarrow^X_Y$

  - **příklad:** $(1, 2, 3, 4, 5) \downarrow^{(A,B,C,D,E)}_{(D,A,E)} = (4, 1, 5)$

- **Kombinace** $c = c_1 \otimes c_2$ $\quad c_1 = (def_1, con_1)$ and $c_2 = (def_2, con_2)$

  $c = (def, con)$, $con = con_1 \cup con_2$,

$$def(\vec{t}) = def_1(\vec{t} \downarrow^{con}_{con_1}) \times def_2(\vec{t} \downarrow^{con}_{con_2})$$

Definice omezení nad polookruhy

- **Systém** (S, D, V):

  S c-polookruh, D konečná doména, V uspořádaná množina proměnných

- **Soft omezení** (def, con): rozsah omezení $con \subseteq V$, $def : D^{|con|} \rightarrow \mathcal{A}$

- **Soft problém** je (C, con) nad (S, D, V), kde $con \subseteq V$, C množina omezení

- **Projekce n-tic** $\vec{t} \downarrow^X_Y$

  - **příklad:** $(1, 2, 3, 4, 5) \downarrow^{(A,B,C,D,E)}_{(D,A,E)} = (4, 1, 5)$

- **Kombinace** $c = c_1 \otimes c_2$ $\qquad c_1 = (def_1, con_1)$ and $c_2 = (def_2, con_2)$

  $c = (def, con)$, $con = con_1 \cup con_2$,

$$def(\vec{t}) = def_1(\vec{t} \downarrow^{con}_{con_1}) \times def_2(\vec{t} \downarrow^{con}_{con_2})$$

  - **příklad:** CSP s váhami: $\mathcal{A} = \mathbb{N} \cup \{\infty\}$, $+ \equiv \min$, $\times \equiv +$, $+\infty$, 0

    zadáno omezení c_1 na xy: aa 2, ab 4, ba 1, bb 0,

    zadáno omezení c_2 na x: a 0, b 1

    kombinace $c = c_1 \otimes c_2$:

Definice omezení nad polookruhy

- **Systém** (S, D, V):
  S c-polookruh, D konečná doména, V uspořádaná množina proměnných
- **Soft omezení** (def, con): rozsah omezení $con \subseteq V$, $def : D^{|con|} \rightarrow \mathcal{A}$
- **Soft problém** je (C, con) nad (S, D, V), kde $con \subseteq V$, C množina omezení
- **Projekce n-tic** $\vec{t} \downarrow^X_Y$
  - **příklad:** $(1, 2, 3, 4, 5) \downarrow^{(A,B,C,D,E)}_{(D,A,E)} = (4, 1, 5)$
- **Kombinace** $c = c_1 \otimes c_2$ $\quad$ $c_1 = (def_1, con_1)$ and $c_2 = (def_2, con_2)$
  $c = (def, con)$, $con = con_1 \cup con_2$,
  $$def(\vec{t}) = def_1(\vec{t} \downarrow^{con}_{con_1}) \times def_2(\vec{t} \downarrow^{con}_{con_2})$$
  - **příklad:** CSP s váhami: $\mathcal{A} = \mathbb{N} \cup \{\infty\}$, $+ \equiv \min$, $\times \equiv +$, $+\infty$, 0
    zadáno omezení c_1 na xy: aa 2, ab 4, ba 1, bb 0,
    zadáno omezení c_2 na x: a 0, b 1
    kombinace $c = c_1 \otimes c_2$: aa 2(=2+0) ab 4(=4+0) ba 2(=1+1) bb 1(=0+1)

Řešení pro omezení nad polookruhy

- **Projekce** $c \Downarrow_I$ $\quad c = (def, con),\ I \subseteq V\ c' = (def', con'),\ con' = con \cap I$

$$def'(\vec{t}') = \sum_{\vec{t}/\vec{t}\downarrow^{con}_{I \cap con} = \vec{t}'} def(\vec{t})$$

  - **příklad (pokračování):** c_1 na xy: aa 2, ab 4, ba 1, bb 0, projekce $c_1 \Downarrow_{\{x\}}$:

Řešení pro omezení nad polookruhy

- **Projekce $c \Downarrow_I$** $\quad c = (def, con)$, $I \subseteq V$ $c' = (def', con')$, $con' = con \cap I$

$$def'(\vec{t}') = \sum_{\vec{t}/\vec{t}\downarrow^{con}_{I \cap con} = \vec{t}'} def(\vec{t})$$

  - **příklad (pokračování):** c_1 na xy: aa 2, ab 4, ba 1, bb 0, $\quad$ projekce $c_1 \Downarrow_{\{x\}}$: a 2, b 0

Řešení pro omezení nad polookruhy

- **Projekce $c \Downarrow_I$** $\quad c = (def, con),\ I \subseteq V\ \ c' = (def', con'),\ con' = con \cap I$

$$def'(\vec{t}') = \sum_{\vec{t}/\vec{t}\downarrow^{con}_{I\cap con}=\vec{t}'} def(\vec{t})$$

  - příklad (pokračování): c_1 na xy: aa 2, ab 4, ba 1, bb 0, projekce $c_1 \Downarrow_{\{x\}}$: a 2, b 0

- **Úroveň splnění** problému $P = (C, con)$ udává omezení

$$Sol(P) = (\bigotimes C) \Downarrow_{con}$$

  - kombinace všech omezení v C a následovně projekce na proměnné v con
  - pro každé přiřazení proměnných v con

    vrací omezení $Sol(P)$ jeho úroveň splnění
  - příklad (pokračování): c_1 na xy: aa 2, ab 4, ba 1, bb 0 c_2 na x: a 0, b 1

    problém $P_1 = (\{c_1, c_2\}, \{x, y\})$: $Sol(P_1) = (c_1 \otimes c_2) \Downarrow_{\{x,y\}}$:

Řešení pro omezení nad polookruhy

- **Projekce** $c \Downarrow_I$ $\quad c = (def, con),\ I \subseteq V\ \ c' = (def', con'),\ con' = con \cap I$

$$def'(\vec{t}') = \sum_{\vec{t}/\vec{t}\downarrow^{con}_{I\cap con}=\vec{t}'} def(\vec{t})$$

  - příklad (pokračování): c_1 na xy: aa 2, ab 4, ba 1, bb 0, projekce $c_1 \Downarrow_{\{x\}}$: a 2, b 0

- **Úroveň splnění** problému $P = (C, con)$ udává omezení

$$Sol(P) = (\bigotimes C) \Downarrow_{con}$$

  - kombinace všech omezení v C a následovně projekce na proměnné v con
  - pro každé přiřazení proměnných v con

    vrací omezení $Sol(P)$ jeho úroveň splnění
  - příklad (pokračování): c_1 na xy: aa 2, ab 4, ba 1, bb 0 $\quad c_2$ na x: a 0, b 1

    problém $P_1 = (\{c_1, c_2\}, \{x, y\})$: $\quad Sol(P_1) = (c_1 \otimes c_2) \Downarrow_{\{x,y\}}$: aa 2, ab 4, ba 2, bb 1

Řešení pro omezení nad polookruhy

- **Projekce** $c \Downarrow_I$ $\quad c = (def, con),\ I \subseteq V\ c' = (def', con'),\ con' = con \cap I$

$$def'(\vec{t'}) = \sum_{\vec{t}/\vec{t}\downarrow^{con}_{I \cap con} = \vec{t'}} def(\vec{t})$$

  - **příklad (pokračování):** c_1 na xy: aa 2, ab 4, ba 1, bb 0, projekce $c_1 \Downarrow_{\{x\}}$: a 2, b 0

- **Úroveň splnění** problému $P = (C, con)$ udává omezení

$$Sol(P) = (\bigotimes C) \Downarrow_{con}$$

  - kombinace všech omezení v C a následovně projekce na proměnné v con
  - pro každé přiřazení proměnných v con

    vrací omezení $Sol(P)$ jeho úroveň splnění
  - **příklad (pokračování):** c_1 na xy: aa 2, ab 4, ba 1, bb 0 c_2 na x: a 0, b 1

    problém $P_1 = (\{c_1, c_2\}, \{x, y\})$: $Sol(P_1) = (c_1 \otimes c_2) \Downarrow_{\{x,y\}}$: aa 2, ab 4, ba 2, bb 1

    problém $P_2 = (\{c_1, c_2\}, \{x\})$: $Sol(P_2) = (c_1 \otimes c_2) \Downarrow_{\{x\}}$:

Řešení pro omezení nad polookruhy

- **Projekce** $c \Downarrow_I$ $\quad c = (def, con),\ I \subseteq V\ \ c' = (def', con'),\ con' = con \cap I$

$$def'(\vec{t}') = \sum_{\vec{t}/\vec{t}\downarrow^{con}_{I\cap con}=\vec{t}'} def(\vec{t})$$

  - příklad (pokračování): c_1 na xy: aa 2, ab 4, ba 1, bb 0, projekce $c_1 \Downarrow_{\{x\}}$: a 2, b 0

- **Úroveň splnění** problému $P = (C, con)$ udává omezení

$$Sol(P) = (\bigotimes C) \Downarrow_{con}$$

  - kombinace všech omezení v C a následovně projekce na proměnné v con
  - pro každé přiřazení proměnných v con

    vrací omezení $Sol(P)$ jeho úroveň splnění
  - příklad (pokračování): c_1 na xy: aa 2, ab 4, ba 1, bb 0 $\quad c_2$ na x: a 0, b 1

    problém $P_1 = (\{c_1, c_2\}, \{x, y\})$: $\quad Sol(P_1) = (c_1 \otimes c_2) \Downarrow_{\{x,y\}}$: aa 2, ab 4, ba 2, bb 1

    problém $P_2 = (\{c_1, c_2\}, \{x\})$: $\quad Sol(P_2) = (c_1 \otimes c_2) \Downarrow_{\{x\}}$: a 2, b 1

Řešení pro omezení nad polookruhy

- **Projekce** $c \Downarrow_I$ $\qquad c = (def, con),\ I \subseteq V\ \ c' = (def', con'),\ con' = con \cap I$

$$def'(\vec{t}') = \sum_{\vec{t}/\vec{t}\downarrow^{con}_{I\cap con}=\vec{t}'} def(\vec{t})$$

  - **příklad (pokračování):** c_1 na xy: aa 2, ab 4, ba 1, bb 0, $\qquad$ projekce $c_1 \Downarrow_{\{x\}}$: a 2, b 0

- **Úroveň splnění** problému $P = (C, con)$ udává omezení

$$Sol(P) = (\bigotimes C) \Downarrow_{con}$$

  - kombinace všech omezení v C a následovně projekce na proměnné v con
  - pro každé přiřazení proměnných v con

    vrací omezení $Sol(P)$ jeho úroveň splnění
  - **příklad (pokračování):** c_1 na xy: aa 2, ab 4, ba 1, bb 0 $\qquad c_2$ na x: $\quad$ a 0, b 1

    problém $P_1 = (\{c_1, c_2\}, \{x, y\})$: $\quad Sol(P_1) = (c_1 \otimes c_2) \Downarrow_{\{x,y\}}$: aa 2, ab 4, ba 2, bb 1

    problém $P_2 = (\{c_1, c_2\}, \{x\})$: $\quad Sol(P_2) = (c_1 \otimes c_2) \Downarrow_{\{x\}}$: a 2, b 1

- **Úroveň konzistence:** $blevel(P) = Sol(P) \Downarrow_{\emptyset}$
  - **příklad (pokračování):** $blevel(P_1) = blevel(P_2) = 1$

Optimalizace & soft omezení: algoritmy

Soft propagace

- Klasická propagace: eliminace nekonzistentních hodnot z domén proměnných
- **Soft propagace**: propagace preferencí (cen) nad k-ticemi hodnot proměnných
  - snaha o získání realističtějších preferencí
  - výpočet realističtějších příspěvků pro cenovou funkci

Soft propagace

- Klasická propagace: eliminace nekonzistentních hodnot z domén proměnných
- **Soft propagace**: propagace preferencí (cen) nad k-ticemi hodnot proměnných
  - snaha o získání realističtějších preferencí
  - výpočet realističtějších příspěvků pro cenovou funkci
- C je **soft k-konzistentní**, jestliže pro všechny podmnožiny $(k-1)$ proměnných W a libovolnou další proměnnou y platí

$$\otimes\{c_i \mid c_i \in C \wedge con_i \subseteq W\} = (\otimes\{c_i \mid c_i \in C \wedge con_i \subseteq (W \cup \{y\})\}) \Downarrow_W$$

  - con_i označuje proměnné v omezení c_i
  - uvažování (def) pouze omezení ve W stejné jako:
    uvažování (def) omezení ve W a omezení spojující y s W, s následnou projekcí na W

Soft hranová konzistence (SAC)

- $k = 2, W = \{x\} : c_x = (\otimes\{c_y, c_{xy}, c_x\}) \Downarrow_{\{x\}}$

Soft hranová konzistence (SAC)

- $k = 2, W = \{x\} : c_x = (\otimes\{c_y, c_{xy}, c_x\}) \Downarrow_{\{x\}}$
- **CSP:** libovolná hodnota v doméně x může být rozšířena o hodnotu v doméně y tak, že je c_{xy} splněno
  - hodnoty a v doméně x, které nelze rozšířit na y, mají $def((a)) = \mathbf{0}$

Soft hranová konzistence (SAC)

- $k = 2, W = \{x\} : c_x = (\otimes\{c_y, c_{xy}, c_x\}) \Downarrow_{\{x\}}$
- **CSP:** libovolná hodnota v doméně x může být rozšířena o hodnotu v doméně y tak, že je c_{xy} splněno
  - hodnoty a v doméně x, které nelze rozšířit na y, mají $def((a)) = \mathbf{0}$
- **Fuzzy CSP:** úroveň preference všech hodnot x v c_x je stejná jako úroveň preference daná $(\otimes\{c_y, c_{xy}, c_x\}) \Downarrow_{\{x\}}$

Soft hranová konzistence (SAC)

- $k = 2, W = \{x\} : c_x = (\otimes\{c_y, c_{xy}, c_x\}) \Downarrow_{\{x\}}$
- CSP: libovolná hodnota v doméně x může být rozšířena o hodnotu v doméně y tak, že je c_{xy} splněno
  - hodnoty a v doméně x, které nelze rozšířit na y, mají $def((a)) = \mathbf{0}$
- Fuzzy CSP: úroveň preference všech hodnot x v c_x je stejná jako úroveň preference daná $(\otimes\{c_y, c_{xy}, c_x\}) \Downarrow_{\{x\}}$
- Příklad na fuzzy CSP: $\mathcal{A} = \langle 0, 1\rangle$, $+ \equiv \max$, $\times \equiv \min$, 0, 1
  - $x, y \in \{a, b\}$
  - $c_x : a \ldots 0.9, b \ldots 0.1$, $c_y : a \ldots 0.9, b \ldots 0.5$, $c_{xy} : aa \ldots 0.8$, $ab \ldots 0.2$, $ba \ldots 0$, $bb \ldots 0$
  - není SAC: c_x dává 0.9 na $x = a$, ale $(\otimes\{c_y, c_{xy}, c_x\}) \Downarrow_{\{x\}}$ dává 0.8 na $x = a$

    $\otimes\{c_y, c_{xy}, c_x\}$ dává na (a, a) :

    $\otimes\{c_y, c_{xy}, c_x\}$ dává na (a, b) :

    projekce $(\otimes\{c_y, c_{xy}, c_x\}) \Downarrow_{\{x\}}$ dává na (a) :

Soft hranová konzistence (SAC)

- $k = 2, W = \{x\} : c_x = (\otimes\{c_y, c_{xy}, c_x\}) \Downarrow_{\{x\}}$
- CSP: libovolná hodnota v doméně x může být rozšířena o hodnotu v doméně y tak, že je c_{xy} splněno
  - hodnoty a v doméně x, které nelze rozšířit na y, mají $def((a)) = \mathbf{0}$
- Fuzzy CSP: úroveň preference všech hodnot x v c_x je stejná jako úroveň preference daná $(\otimes\{c_y, c_{xy}, c_x\}) \Downarrow_{\{x\}}$
- Příklad na fuzzy CSP: $\mathcal{A} = \langle 0, 1\rangle$, $+ \equiv \max$, $\times \equiv \min$, 0, 1
  - $x, y \in \{a, b\}$
  - $c_x : a \ldots 0.9, b \ldots 0.1$, $c_y : a \ldots 0.9, b \ldots 0.5$, $c_{xy} : aa \ldots 0.8$, $ab \ldots 0.2$, $ba \ldots 0$, $bb \ldots 0$
  - není SAC: c_x dává 0.9 na $x = a$, ale $(\otimes\{c_y, c_{xy}, c_x\}) \Downarrow_{\{x\}}$ dává 0.8 na $x = a$
    $\otimes\{c_y, c_{xy}, c_x\}$ dává na $(a, a) : \min(0.9, 0.8, 0.9) = 0.8$
    $\otimes\{c_y, c_{xy}, c_x\}$ dává na $(a, b) : \min(0.5, 0.2, 0.9) = 0.2$
    projekce $(\otimes\{c_y, c_{xy}, c_x\}) \Downarrow_{\{x\}}$ dává na $(a) : \max(0.8, 0.2) = 0.8$

Výpočet SAC

- Základní algoritmus pro **výpočet SAC:**
  - pro každé x a y změnit definici c_x tak, aby korespondovala s $(\otimes\{c_y, c_{xy}, c_x\}) \Downarrow_{\{x\}}$
  - iterace až do dosažení stability

Výpočet SAC

- Základní algoritmus pro **výpočet SAC:**
  - pro každé x a y změnit definici c_x tak, aby korespondovala s $(\otimes\{c_y, c_{xy}, c_x\}) \Downarrow_{\{x\}}$
  - iterace až do dosažení stability
- $\times$ idempotentní (fuzzy CSP)
  - zajištěna ekvivalence, tj. původní i nový (po dosažení SAC) problém mají stejné řešení
  - zajištěno ukončení algoritmu

Výpočet SAC

- Základní algoritmus pro **výpočet SAC:**
  - pro každé x a y změnit definici c_x tak, aby korespondovala s $(\otimes\{c_y, c_{xy}, c_x\}) \Downarrow_{\{x\}}$
  - iterace až do dosažení stability
- $\times$ idempotentní (fuzzy CSP)
  - zajištěna ekvivalence, tj. původní i nový (po dosažení SAC) problém mají stejné řešení
  - zajištěno ukončení algoritmu
- $\times$ není idempotentní (CSP s váhami)
  - pro každé $u, v \in \mathcal{A}$ existuje $w \in \mathcal{A}$ taková, že $v \times w = u$
    - w se nazývá **rozdíl** mezi u a v, **maximální rozdíl** se značí $u - v$
    - nutno požadovat novou vlastnost pro systém (a rozšířit projekci a kombinaci)
  - s novou vlastností zajištěna ekvivalence, při striktní monotonii zajištěno i ukončení
    - tato vlastnost platí pro CSP s váhami

Řešení COP

- **Cíl:** nalezení úplného řešení s optimální hodnotou cenové funkce
- **Prohledávání**
  - lokální prohledávání
    - přímé zahrnutí optimalizačního kriteria do evaluace (hodnota obj. funkce)
  - stromové prohledávání
    - metoda větví a mezí + její rozšíření

Řešení COP

- **Cíl:** nalezení úplného řešení s optimální hodnotou cenové funkce
- **Prohledávání**
  - lokální prohledávání
    - přímé zahrnutí optimalizačního kriteria do evaluace (hodnota obj. funkce)
  - stromové prohledávání
    - metoda větví a mezí + její rozšíření
- CSP s omezeními: **minimalizace součtu vah omezení** $\min \sum_{c \in C} c(\vec{d})$
  - velmi častá optimalizační úloha
  - váha (**cena**) omezení: 0 = úplné splnění, $(0, \infty)$ částečné nesplnění, ∞ úplné nesplnění
  - hodnota cenové funkce: **cena přiřazení/řešení**
  - maximalizace je duální problém
  - algoritmy pro fuzzy CSP na podobných principech

COP jako série CSP problémů

- Triviální metoda řešení

- Řešení COP jako CSP a nalezení iniciální hodnoty cenové funkce W^1
- **Opakované řešení CSP ($i = 1 \ldots$)**
  s přidáním omezení $\sum_{c \in C} c(\vec{d}) < W^i$
- Když řešení nového CSP neexistuje, pak poslední W^i dává optimum

COP jako série CSP problémů

- Triviální metoda řešení
- Řešení COP jako CSP a nalezení iniciální hodnoty cenové funkce W^1
- **Opakované řešení CSP ($i = 1 \ldots$)**
  s přidáním omezení $\sum_{c \in C} c(\vec{d}) < W^i$
- Když řešení nového CSP neexistuje, pak poslední W^i dává optimum
- Výpočetně zbytečně náročné
- Efektivní rozšíření obtížné

Metoda větví a mezí (*branch&bound*) BB

- Prohledávání stromu do hloubky
  - přiřazené=**minulé proměnné** P, nepřiřazené=**budoucí proměnné** F
  - omezení pouze na minulých proměnných C_P, omezení na minulých i budoucích proměnných C_{PF}, omezení pouze na budoucích proměnných C_F

Metoda větví a mezí (*branch&bound*) BB

- Prohledávání stromu do hloubky
  - přiřazené=**minulé proměnné** P, nepřiřazené=**budoucí proměnné** F
  - omezení pouze na minulých proměnných C_P, omezení na minulých i budoucích proměnných C_{PF}, omezení pouze na budoucích proměnných C_F
- Výpočet **mezí**
  - **horní mez** UB: cena nejlepšího dosud nalezeného řešení
  - **dolní mez** LB: dolní odhad minimální ceny pro současné částečné přiřazení

Metoda větví a mezí (*branch&bound*) BB

- Prohledávání stromu do hloubky
  - přiřazené=**minulé proměnné** P, nepřiřazené=**budoucí proměnné** F
  - omezení pouze na minulých proměnných C_P, omezení na minulých i budoucích proměnných C_{PF}, omezení pouze na budoucích proměnných C_F
- Výpočet **mezí**
  - **horní mez** UB: cena nejlepšího dosud nalezeného řešení
  - **dolní mez** LB: dolní odhad minimální ceny pro současné částečné přiřazení
- **Ořezávání**: $LB \geq UB$ (cíl: minimalizace)
  - víme, že rozšíření současného částečného řešení už bude mít horší (vyšší) cenu LB než dosud nalezené řešení UB
  - lze proto ořezat tuto část prohledávacího prostoru
  - příklad: pokud nalezneme řešení s cenou 10 odřízneme všechny větve, které mají cenu vyšší než 9

Metoda větví a mezí: výběr hodnoty

- **Algorimus metody větví a mezí**
  - generický algoritmus rozšiřitelný jako implementace backtrackingu
  - možná rozšíření zejména o: pohled dopředu, výpočet dolní meze
- $LB(\vec{d})$ vrací dolní odhad ceny pro každé částečné přiřazení $\vec{d}$
  - použití při rozšíření o jednu proměnnou $LB(\vec{a}_{i-1}, a)$

Metoda větví a mezí: výběr hodnoty

- **Algorimus metody větví a mezí**
  - generický algoritmus rozšiřitelný jako implementace backtrackingu
  - možná rozšíření zejména o: pohled dopředu, výpočet dolní meze
- $LB(\vec{d})$ vrací dolní odhad ceny pro každé částečné přiřazení $\vec{d}$
  - použití při rozšíření o jednu proměnnou $LB(\vec{a}_{i-1}, a)$
- **Optimistický výběr hodnoty**

```
procedure Select-Value-BB
while D'_i is not empty
      vyber a smaž libovolný a ∈ D'_i takový, že min LB(a⃗_{i-1}, a)
      if Consistent(a⃗_{i-1}, x_i = a) a LB(a⃗_{i-1}, a) < UB
         return a                          (jinak ořezej a)
return null                                (konzistentní hodnota neexistuje)
```

Algoritmus metody větví a mezí

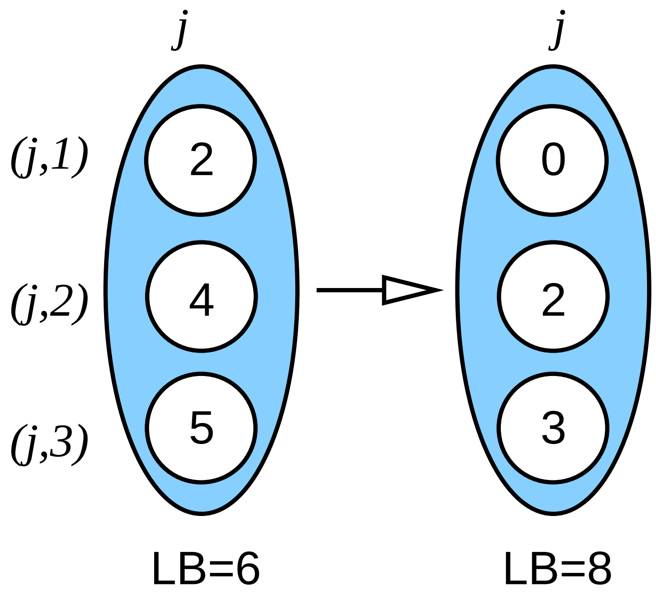
```
procedure Branch-Bound((X,D,C),UB,f)                          rozdíly od backtrackingu
i := 1, D'_i := D_i            (inicializace čítače proměnných, kopírování domény)
Reseni := null                   (řešení dosud nenalezeno)
repeat while 1 ≤ i ≤ n
            přiřazení x_i := Select-Value-BB
            if x_i is null          (žádná hodnota nebyla vrácena)
               i := i − 1           (zpětná fáze)
            else i := i + 1         (dopředná fáze)
                 D'_i := D_i
      if i = 0  if Reseni is not null
                  return UB, Reseni
                else return ,,nekonzistentní''
      else Reseni := x_1,...,x_n    (uložení hodnot dosud nejlepšího přiřazení)
           spočítej cenu současného přiřazení W = Σ_{c∈C} c(x⃗), UB:=min{W, UB}
           i := 1, nastav D'_k na D_k pro k = 1 ... n
until true
end Branch-Bound
```

Dolní mez

- **Kvalita dolní meze**: velmi důležitá pro efektivitu BB
- Dolní mez lze ovlivnit pomocí
  - **ceny minulých proměnných**
    - vzdálenost (součet vah omezení na minulých proměnných)
  - **lokální ceny budoucích proměnných vzhledem k minulým proměnným**
    - NC*
  - **lokální ceny budoucích proměnných**
    - AC*
  - **globální ceny budoucích proměnných**
    - prohledávání ruská panenka

NC* algoritmus

- Projekce ceny hodnot $(j, *)$ pro každou proměnnou j do dolní hranice LB ceny řešení

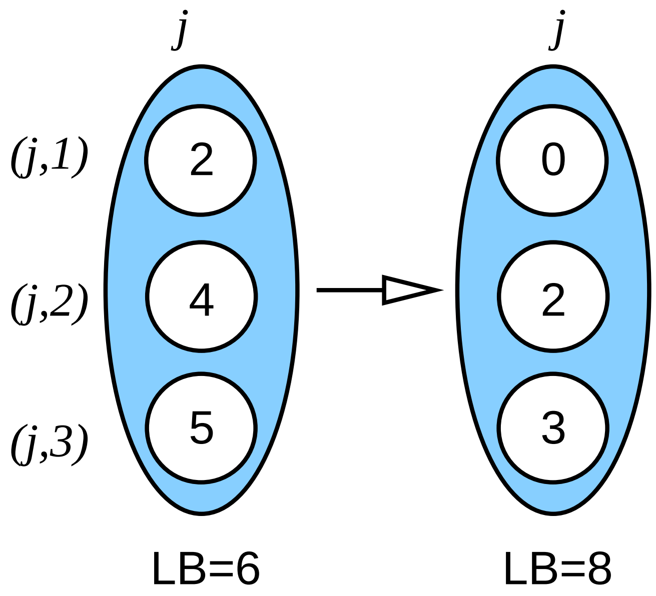


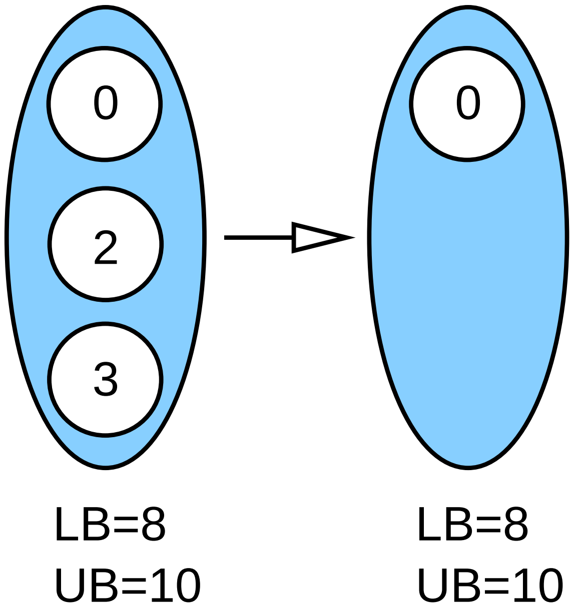
- Hodnotu a proměnné j značíme (j, a)
  - obrázek: proměnná j má nejprve tři hodnoty $(j, 1), (j, 2), (j, 3)$, jejichž cena je 2,4 a 5
- Všechny hodnoty proměnné j značíme $(j, *)$

NC* algoritmus

- Projekce ceny hodnot $(j,*)$ pro každou proměnnou j do dolní hranice LB ceny řešení

- Smazání hodnot (j,a) převyšující (nebo rovné) horní hranici UB



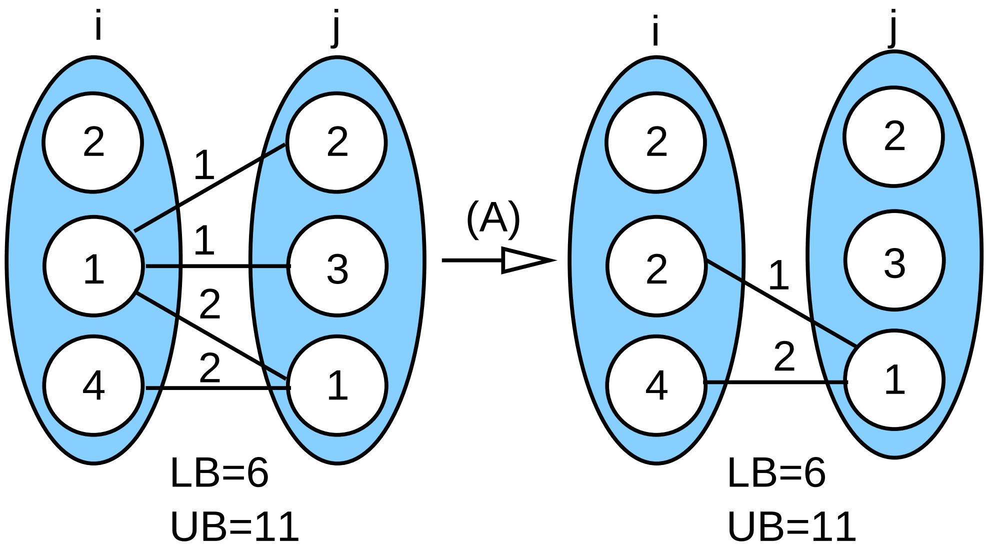


- Hodnotu a proměnné j značíme (j,a)
  - obrázek: proměnná j má nejprve tři hodnoty $(j,1),(j,2),(j,3)$, jejichž cena je 2,4 a 5

- Všechny hodnoty proměnné j značíme $(j,*)$

AC* algoritmus

(A) Projekce ceny hrany $(i, a)(j, b)$ do ceny hodnoty (i, a), pokud je tato cena zahrnuta ve všech hranách $(i, a)(j, *)$



$(i, 2)(j, 1) \quad \longrightarrow$

$(i, 2)(j, 2) \quad \longrightarrow \quad (i, 2)$

$(i, 2)(j, 3) \quad \longrightarrow$

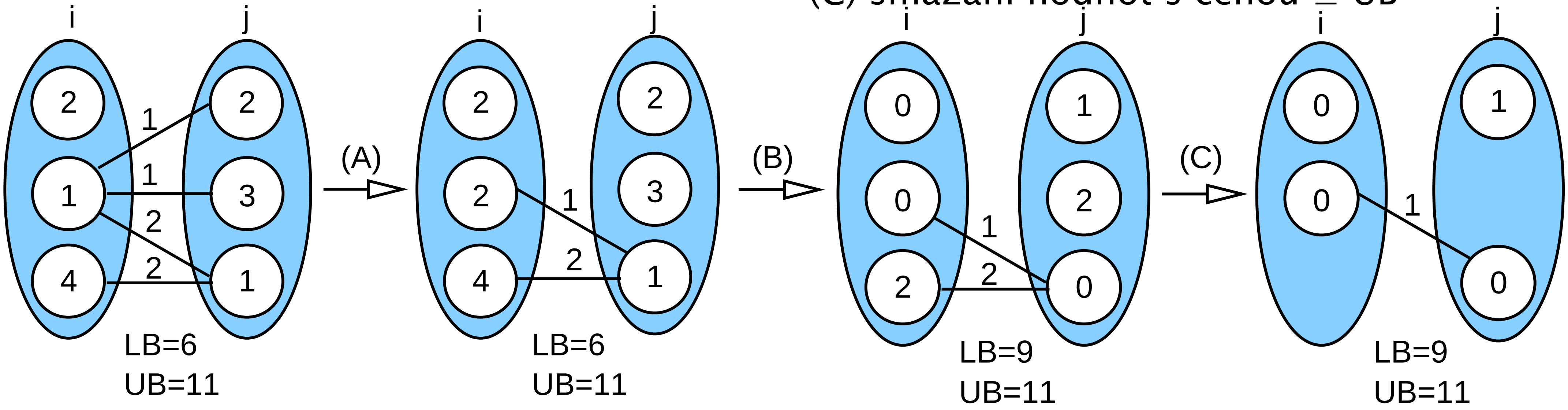
AC* algoritmus

(A) Projekce ceny hrany $(i,a)(j,b)$ do ceny hodnoty (i,a), pokud je tato cena zahrnuta ve všech hranách $(i,a)(j,*)$

- NC* algoritmus

  (B) projekce ceny hodnot pro každou proměnnou

  (C) smazání hodnot s cenou $\geq$ UB



$(i,2)(j,1) \quad \longrightarrow$

$(i,2)(j,2) \quad \longrightarrow \quad (i,2)$

$(i,2)(j,3) \quad \longrightarrow$

Prohledávání ruská panenka (*Russion doll search*)

- **n po sobě jdoucích BB prohledávání**, každé má navíc jednu proměnnou

  statické uspořádání proměnných

  - první podproblém obsahuje pouze n-tou proměnnou

    i-tý podproblém obsahuje posledních i proměnných $((n - i + 1) \dots n)$
  - podproblémy řešeny pomocí BB s využitím LB a UB z předchozích běhů

Prohledávání ruská panenka (*Russion doll search*)

- **n po sobě jdoucích BB prohledávání**, každé má navíc jednu proměnnou

  statické uspořádání proměnných

  - první podproblém obsahuje pouze n-tou proměnnou

    i-tý podproblém obsahuje posledních i proměnných $((n-i+1)\dots n)$
  - podproblémy řešeny pomocí BB s využitím LB a UB z předchozích běhů
- Při **řešení podproblému** $(n-i+1)$
  - problém zahrnuje proměnné $x_i, x_{i+1}, \dots, x_n$
  - mějme částečné přiřazení pro tento podproblém $(a_i, a_{i+1}, \dots, a_{i+j})$ s nepřiřazenými proměnnými $x_{i+j+1}, \dots, x_n$

Prohledávání ruská panenka (*Russion doll search*)

- n **po sobě jdoucích BB prohledávání**, každé má navíc jednu proměnnou
  statické uspořádání proměnných
  - první podproblém obsahuje pouze n-tou proměnnou
    i-tý podproblém obsahuje posledních i proměnných $((n-i+1)\dots n)$
  - podproblémy řešeny pomocí BB s využitím LB a UB z předchozích běhů
- Při **řešení podproblému** $(n-i+1)$
  - problém zahrnuje proměnné $x_i, x_{i+1}, \dots, x_n$
  - mějme částečné přiřazení pro tento podproblém $(a_i, a_{i+1}, \dots, a_{i+j})$ s nepřiřazenými proměnnými $x_{i+j+1}, \dots, x_n$
  - do dolní meze lze zahrnout optimální cenu $(n-i-j)$ podproblému $optim(x_{i+j+1}, \dots, x_n)$
    - $LB((a_i, a_{i+1}, \dots, a_{i+j})) = dist((a_i, a_{i+1}, \dots, a_{i+j})) + optim(x_{i+j+1}, \dots, x_n)$
      $dist((a_i, a_{i+1}, \dots, a_{i+j}))$ vzdálenost (součet vah omezení na minulých proměnných)
  - optimální řešení přechozích problémů použita pro:
    výběr hodnoty, pro zlepšení iniciální horní meze

Prohledávání ruská panenka (*Russion doll search*)

- **n po sobě jdoucích BB prohledávání**, každé má navíc jednu proměnnou
  - první podproblém obsahuje pouze n-tou proměnnou
    statické uspořádání proměnných

    i-tý podproblém obsahuje posledních i proměnných $((n - i + 1) \dots n)$
  - podproblémy řešeny pomocí BB s využitím LB a UB z předchozích běhů
- Při **řešení podproblému** $(n - i + 1)$
  - problém zahrnuje proměnné $x_i, x_{i+1}, \dots, x_n$
  - mějme částečné přiřazení pro tento podproblém $(a_i, a_{i+1}, \dots, a_{i+j})$ s nepřiřazenými proměnnými $x_{i+j+1}, \dots, x_n$
  - do dolní meze lze zahrnout optimální cenu $(n - i - j)$ podproblému $optim(x_{i+j+1}, \dots, x_n)$
    - $LB((a_i, a_{i+1}, \dots, a_{i+j})) = dist((a_i, a_{i+1}, \dots, a_{i+j})) + optim(x_{i+j+1}, \dots, x_n)$

      $dist((a_i, a_{i+1}, \dots, a_{i+j}))$ vzdálenost (součet vah omezení na minulých proměnných)
  - optimální řešení přechozích problémů použita pro:

    výběr hodnoty, pro zlepšení iniciální horní meze
- Vnořená prohledávání se vyplatí vzhledem k prořezání stavového prostoru