Cvičení 7

Programování
s omezujícími podmínkami

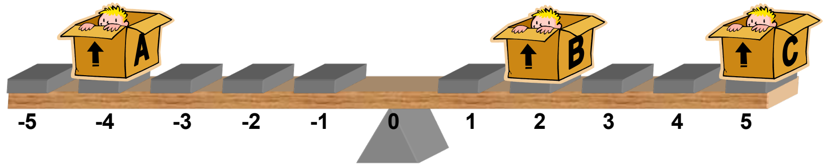
Roman Barták
Katedra teoretické informatiky a matematické logiky

roman.bartak@mff.cuni.cz
http://ktiml.mff.cuni.cz/~bartak

Pohoupejte nás

Zadání:

Adam (36 kg), Boris (32 kg) a Cecil (16 kg) se chtějí usadit na houpačku s deseti sedátky tak, aby houpačka byla vyvážená a měli mezi sebou vždy alespoň dvě sedačky.



CSP model:

A,B,C in -5..5	pozice na houpačce
36*A+32*B+16*C = 0	vyvážený stav
|A-B|>2, |A-C|>2, |B-C|>2	minimální vzdálenosti

Pohoupejte nás
implementace

```
:-use_module(library(clpfd)).

seesaw(Sol):-
    Sol = [A,B,C],

    domain(Sol,-5,5),
    36*A+32*B+16*C #= 0,
    abs(A-B)#>2, abs(A-C)#>2, abs(B-C)#>2,

    labeling([ff],Sol).
```

```
?- seesaw(X).

X = [-4,2,5] ? ;
X = [-4,4,1] ? ;
X = [-4,5,-1] ? ;
X = [4,-5,1] ? ;
X = [4,-4,-1] ? ;
X = [4,-2,-5] ? ;

no
```

Odstranění symetrií

velice důležité pro zmenšení prohledávaného prostoru

```
:-use_module(library(clpfd)).

seesaw(Sol):-
    Sol = [A,B,C],

    domain(Sol,-5,5),
    A #< 0,
    36*A+32*B+16*C #= 0,
    abs(A-B)#>2, abs(A-C)#>2, abs(B-C)#>2,

    labeling([ff],Sol).
```

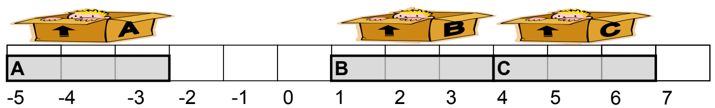
```
?- seesaw(X).

X = [-4,2,5] ? ;
X = [-4,4,1] ? ;
X = [-4,5,-1] ? ;

no
```

Pohoupejte nás
z jiného pohledu

```
domain([A,B,C],-5,5),
A #< 0,
36*A+32*B+16*C #= 0,
abs(A-B)#>2,
abs(A-C)#>2,
abs(B-C)#>2
```

```
A in -4..0
B in -1..5
C in -5..5
```

Množina podobných podmínek často indikuje podproblém, který může být reprezentován pomocí **globální podmínky**.



Můžeme používat globální podmínky popisující alokaci aktivit na unární zdroj.

```
domain([A,B,C],-5,5),
A #< 0,
36*A+32*B+16*C #= 0,
serialized([A,B,C],[3,3,3],[])
```

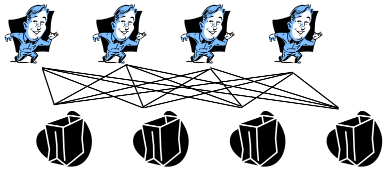
startovní čas — trvání — precedence

```
A in -4..0
B in -1..5
C in (-5..-3) \/ (-1..5)
```

Přiřazovací problém

Zadání:

Přiřaďte čtyři pracovníky na výrobu čtyř produktů tak, že každý pracovník dělá jeden produkt a každý produkt je vyráběn jedním pracovníkem. Efektivita výroby je dána následující tabulkou a celková efektivita musí být minimálně 19.



	P1	P2	P3	P4
W1	7	1	3	4
W2	8	2	5	1
W3	4	3	7	2
W4	3	1	6	3

CSP model:

W1,W2,W3,W4 in 1..4 — produkt pro pracovníka

all_different([W1,W2,W3,W4]) — různé produkty

$T_{1,W1}+T_{2,W2}+T_{3,W3}+T_{4,W4} \geq 19$ — celková efektivita

Přiřazovací problém
implementace

```
:-use_module(library(clpfd)).

allocation(Sol):-
    Sol = [W1,W2,W3,W4],

    domain(Sol,1,4),
    all_different(Sol),
    element(W1,[7,1,3,4],EW1),
    element(W2,[8,2,5,1],EW2),
    element(W3,[4,3,7,2],EW3),
    element(W4,[3,2,6,3],EW4),
    EW1+EW2+EW3+EW4 #>= 19,

    labeling([ff],Sol).
```

```
?- allocation(X).

X = [1,2,3,4] ? ;    19
X = [2,1,3,4] ? ;    19
X = [4,1,2,3] ? ;    21
X = [4,1,3,2] ? ;    21

no
```

Optimalizace pomocí B&B

```
    EW1+EW2+EW3+EW4 #= E,

    maximize(labeling([ff],Sol),E).
```

```
?- allocation(X).

X = [4,1,2,3] ? ;

no
```

Jak to funguje?

- najdi první korektní ohodnocení proměnných
- najdi lepší ohodnocení proměnných
- opakuj dokud existuje nějaké ohodnocení

Přiřazovací problém
duální model

Proč přiřazujeme produkt pracovníkům?

Nemohli bychom to udělat obráceně, tj. přiřazovat pracovníky na produkty?

Samozřejmě, že můžeme **prohodit roli proměnných a hodnot**!

Získaný model se potom nazývá **duální model**.

```
:-use_module(library(clpfd)).

allocation_dual(Products):-
    Products = [P1,P2,P3,P4],

    domain(Products,1,4),
    all_different(Products),
    element(P1,[7,8,4,3],EP1),
    element(P2,[1,2,3,1],EP2),
    element(P3,[3,5,7,6],EP3),
    element(P4,[4,1,2,3],EP4),
    EP1+EP2+EP3+EP4 #>= 19,

    labeling([ff],Products).
```

Počet bodů volby

Primární model 15

Duální model 11

```
P1 in 1..2
P2 in 1..4
P3 in 2..4
P4 in 1..4
```

A jaký model je lepší?

V tomto konkrétním případě propaguje duální model dříve, a proto ho lze považovat za lepší.

Přiřazovací problém
skládání modelů

Abychom získali lepší ořezání domén, můžeme oba modely kombinovat v jednom modelu.

```
:-use_module(library(clpfd)).

allocation_combined(Workers):-
    Workers= [W1,W2,W3,W4],
    domain(Workers,1,4),
    all_different(Workers),
    element(W1,[7,1,3,4],EW1),
    element(W2,[8,2,5,1],EW2),
    element(W3,[4,3,7,2],EW3),
    element(W4,[3,1,6,3],EW4),
    EW1+EW2+EW3+EW4 #>= 19,

    Products = [P1,P2,P3,P4],
    domain(Products,1,4),
    all_different(Products),
    element(P1,[7,8,4,3],EP1),
    element(P2,[1,2,3,1],EP2),
    element(P3,[3,5,7,6],EP3),
    element(P4,[4,1,2,3],EP4),
    EP1+EP2+EP3+EP4 #>= 19,

    assignment(Workers,Products),

    labeling([ff],Workers).
```

Primární model

```
W1 in (1..2)\/{4}
W2 in 1..4
W3 in 2..4
W4 in 2..4
```

Duální model (redundantní)

```
P1 in 1..2
P2 in 1..4
P3 in 2..4
P4 in 1..4
```

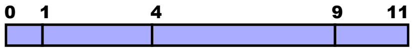
Tunelovací podmínka

Řešení jednoho modelu stačí

Golombovo pravítko

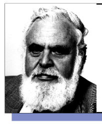
Pravítko s M značkami umístěnými tak, že vzájemné vzdálenosti mezi dvojicemi značek jsou navzájem různé.

Optimální pravítko má nejkratší možnou délku.

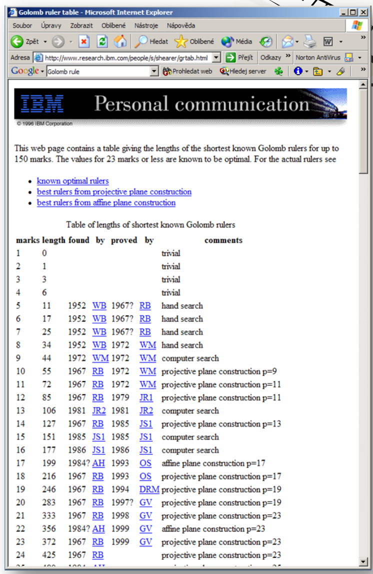


Pro $M \geq 16$ těžké, pro $M \geq 24$ není znám exaktní algoritmus!

Použití v radioastronomii.



Solomon W. Golomb
Professor
University of Southern California
http://csi.usc.edu/faculty/golomb.html



Golomb ruler table - Microsoft Internet Explorer

http://www.research.ibm.com/people/s/shearer/grtab.html

IBM Personal communication

© 1996 IBM Corporation

This web page contains a table giving the lengths of the shortest known Golomb rulers for up to 150 marks. The values for 23 marks or less are known to be optimal. For the actual rulers see

- known optimal rulers
- best rulers from projective plane construction
- best rulers from affine plane construction

Table of lengths of shortest known Golomb rulers

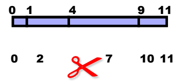
marks	length	found	by	proved	by	comments
1	0					trivial
2	1					trivial
3	3					trivial
4	6					trivial
5	11	1952	WB	1967?	RB	hand search
6	17	1952	WB	1967?	RB	hand search
7	25	1952	WB	1967?	RB	hand search
8	34	1952	WB	1972	WM	hand search
9	44	1972	WM	1972	WM	computer search
10	55	1967	RB	1972	WM	projective plane construction p=9
11	72	1967	RB	1972	WM	projective plane construction p=11
12	85	1967	RB	1979	JR1	projective plane construction p=11
13	106	1981	JR2	1981	JR2	computer search
14	127	1967	RB	1985	JS1	projective plane construction p=13
15	151	1985	JS1	1985	JS1	computer search
16	177	1986	JS1	1986	JS1	computer search
17	199	1984?	AH	1993	OS	affine plane construction p=17
18	216	1967	RB	1993	OS	projective plane construction p=17
19	246	1967	RB	1994	DRM	projective plane construction p=19
20	283	1967	RB	1997?	GV	projective plane construction p=19
21	333	1967	RB	1998	GV	projective plane construction p=23
22	356	1984?	AH	1999	GV	affine plane construction p=23
23	372	1967	RB	1999	GV	projective plane construction p=23
24	425	1967	RB			projective plane construction p=23

Golombovo pravítko
CSP model

Základní model:

Proměnné $X_1, \ldots, X_M$ s doménou 0..M*M	
$X_1 = 0$	začátek pravítka
$X_1 < X_2 < \ldots < X_M$	odstranění permutací
$\forall i<j \; D_{i,j} = X_j - X_i$	proměnné pro rozdíly
all_different($\{D_{1,2}, D_{1,3}, \ldots D_{1,M}, D_{2,3}, \ldots D_{M,M-1}\}$)	

Rozšíření modelu:



$D_{1,2} < D_{M,M-1}$ odstranění symetrie

přesnější meze pro $D_{i,j}$

$D_{i,j} = D_{i,i+1} + D_{i+1,i+2} + \ldots + D_{j-1,j}$

Tedy $D_{i,j} \geq \Sigma_{j-i} = (j-i)*(j-i+1)/2$ dolní mez

$X_M = X_M - X_1 = D_{1,M} = D_{1,2} + D_{2,3} + \ldots D_{i-1,i} + D_{i,j} + D_{j,j+1} + \ldots + D_{M-1,M}$

$D_{i,j} = X_M - (D_{1,2} + \ldots D_{i-1,i} + D_{j,j+1} + \ldots + D_{M-1,M})$

Tedy $D_{i,j} \leq X_M - (M-1-j+i)*(M-j+i)/2$ horní mez

Golombovo pravítko
výsledky

Jaký je efekt různých modelů?

	základní model	základní model + symetrie	základní model + symetrie + meze
7	**220**	**80**	**30**
8	**1 462**	**611**	**190**
9	**13 690**	**5 438**	**1 001**
10	**120 363**	**49 971**	**7 011**
11	**2 480 216**	**985 237**	**170 495**

čas v milisekundách na Mobile Pentium 4-M 1.70 GHz, 768 MB RAM

Jaký je efekt různých prohledávacích strategií?

	fail first			leftmost first		
	enum	**step**	**bisect**	**enum**	**step**	**bisect**
7	**40**	**60**	**40**	**30**	**30**	**30**
8	**390**	**370**	**350**	**220**	**190**	**200**
9	**2 664**	**2 384**	**2 113**	**1 182**	**1 001**	**921**
10	**20 870**	**17 545**	**14 982**	**8 782**	**7 011**	**6 430**
11	**1 004 515**	**906 323**	**779 851**	**209 251**	**170 495**	**159 559**

čas v milisekundách na Mobile Pentium 4-M 1.70 GHz, 768 MB RAM

Poznámky k modelování

- Určení výsledné efektivity různých modelů je těžký úkol, jehož splnění vyžaduje **znalost** chování **řešiče podmínek**.
- Obvykle je nejlepším modelem ten, který **propaguje dříve**.
- Několik **pravidel** pro modelování s podmínkami:
  - **globální podmínky**
    - (+) posilují propagaci při dobré efektivitě
  - **odstranění symetrií**
    - (+) redukuje prohledávaný prostor
  - **redundantní podmínky**
    - (+) posilují propagaci
    - (–) ale přidávají další náklady

    Příklad: duální model